

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Кафедра автоматизованих систем обробки інформації і управління

УДК: 004

«До захисту допущено»

В.о. завідувача кафедри

О.А.Павлов
(ініціали, прізвище)

“ ” 2019 р.

Дипломний проект
на здобуття ступеня бакалавра

з напрямку підготовки 6.050101 «Комп'ютерні науки»

на тему: « Комплекс задач з обліку експонатів картинної галереї »

Виконав:

студент 4 курсу, групи ІС-351

Логвиновський Євгеній Олександрович
(прізвище, ім'я, по батькові)

(підпис)

Керівник

доц., к.т.н., доц. Жаріков Е.В.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

**Консультант з
графічної
документації**

доц., к.т.н., доц. Тєлишева Т.О.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Рецензент

доц., к.т.н., доц. Проценко М.М.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Засвідчую, що у цьому дипломному проекті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент Логвиновський Є.О.

(підпис)

Київ – 2019 р.

Пояснювальна записка до дипломного проекту

на тему: *«Комплекс задач з обліку експонатів картинної галереї»*

Київ – 2019 року

АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка дипломного проекту складається з п'яти розділів, містить 42 рисунки, 12 таблиць, 2 додатки, 14 джерел.

Дипломний проект присвячений розробці комплексу задач з обліку експонатів картинної галереї, що не лише неодноразово переміщуються в її межах, але і тимчасово передаються або продаються назавжди іншим особам.

У розділі інформаційного забезпечення визначено всю початкову інформацію, яка потрапляє до програми та всю звітну інформацію, яку програма видає в результаті роботи. На основі цієї інформації створено структуру БД у 3НФ, яка повністю відповідає комплексу задач.

Розділ математичного забезпечення присвячений використанню мови SQL для формування всіх необхідних звітів з БД.

У розділі програмного та технічного забезпечення за допомогою діаграм UML та словесного опису розглянуто вимоги до технічного забезпечення, обрану клієнт-серверну архітектуру програмного забезпечення, специфікації функцій клієнтської програми та склад звітів. Програмне забезпечення створене мовою с#.

У технологічному розділі наведено керівництво користувача та результати випробування програмного продукту.

ПЕРЕЛІК КЛЮЧОВИХ СЛІВ: ОБЛІК, ЕКСПОНАТ, ПРОГРАМА, КОРИСТУВАЧ, ДОСТУП, ЗВІТ, ФІЛЬТР, БД, СКБД, ТАБЛИЦЯ.

					ДП ІС-5104.1260-с.ПЗ				
		Прізвище	Підпис	Дата					
Розроб.	Логвиновський Є.О.				Комплекс задач з обліку експонатів картинної галереї	Літ.	Лист	Листів	
Перевірів.	Жаріков Е.В.						2	114	
Н. кон.	Тєлишева Т.О.					КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-351			
Затв.	Павлов О.А.								

ABSTRACT

A structure and a scope of the work. The explanatory note of the diploma project consists of five sections. It contains 42 figures, 11 tables, 2 supplements, 14 sources.

The diploma project is devoted to the development of a set of tasks for the accounting of exhibits of the art gallery, which not only repeatedly move within its borders, but also are temporarily transferred or sold forever to other people.

The information section provides all the initial information that comes into the program and all the reporting information that the program issues as a result of work. A database structure has been created in 3NF on the basis of this information. It corresponds to the set of tasks fully.

The section of mathematical support is devoted to using the SQL language to generate all necessary reports from the database.

In the software and hardware section UML diagrams and verbal descriptions are used to settle the technical requirements. Client-server software architecture is selected. Client application specification, and report structure are specified. The software is created in c# programming language.

The technology section provides user guide and test results for the software product.

					ДП ІС-5104.1260-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

ЗМІСТ

ВСТУП.....	6
1 ЗАГАЛЬНІ ПОЛОЖЕННЯ	8
1.1 ОПИС ПРЕДМЕТНОГО СЕРЕДОВИЩА	8
1.1.1 <i>Опис процесу діяльності</i>	8
1.1.2 <i>Опис функціональної моделі</i>	11
1.2 ОГЛЯД НАЯВНИХ АНАЛОГІВ	13
1.3 ПОСТАНОВКА ЗАДАЧІ.....	14
1.3.1 <i>Призначення розробки</i>	14
1.3.2 <i>Цілі та задачі розробки</i>	14
Висновок до розділу	15
2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ	16
2.1 ВХІДНІ ДАНІ	16
2.2 ВИХІДНІ ДАНІ.....	17
2.3 ОПИС СТРУКТУРИ БАЗИ ДАНИХ	20
Висновок до розділу	23
3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ	24
3.1 ЗМІСТОВНА ПОСТАНОВКА ЗАДАЧІ	24
3.2 МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ	24
3.3 ОБГРУНТУВАННЯ МЕТОДУ РОЗВ'ЯЗАННЯ	24
3.4 ОПИС МЕТОДІВ РОЗВ'ЯЗАННЯ	24
Висновок до розділу	26
4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ	27
4.1 ЗАСОБИ РОЗРОБКИ	27
4.2 ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ	27
4.3 АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	27
4.3.1 <i>Діаграма класів</i>	27
4.3.2 <i>Діаграма послідовності</i>	29
4.3.3 <i>Діаграма компонентів</i>	31
4.3.4 <i>Специфікація функцій</i>	33
4.4 ОПИС ЗВІТІВ.....	39

Висновок до розділу	40
5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ	41
5.1 Керівництво користувача	41
5.2 Випробування програмного продукту	58
5.2.1 Мета випробувань	58
5.2.2 Загальні положення	58
5.2.3 Результати випробувань	59
Висновок до розділу	60
ЗАГАЛЬНІ ВИСНОВКИ	61
ПЕРЕЛІК ПОСИЛАНЬ	62
ДОДАТОК А	64
ДОДАТОК Б.....	94

ВСТУП

В процесі діяльності картинної галереї експонати часто переміщуються між різними приміщеннями галереї та за її межі, наприклад, на реставрацію, виїзні експозиції чи тимчасово передаються іншим галереям. Також є потреба обліку експонатів, що викрадені або продані. Стандартні програми складського обліку не враховують всі особливості бізнес-процесів картинної галереї, тому виникла необхідність розробки програми, що врахує специфіку роботи галереї з експонатами.

Також важливою задачею обліку експонатів є збереження історії їх переміщення, що дозволяє персоналу не лише встановити, де знаходиться експонат зараз, але і те, де й коли він перебував у минулому.

Експонатом картинної галереї може бути не лише картина у вузькому розумінні, але й інші вироби, як то художня фотографія, вишивка тощо.

Із загальним переліком експонатів працюють співробітники різних посад, що мають різні задачі згідно свої посадових обов'язків. Доступ стороннім особам до службової інформації не надається.

На основі аналізу предметної області прийнято рішення про створення клієнт-серверного додатку в локальній обчислювальній мережі (ЛОМ) зі стандартним віконним інтерфейсом та розмежуванням доступу користувачів згідно ролей.

В програмі не обмежено кількість та розміри файлів фотографій, пов'язаних з певним експонатом. В тому числі можливе зберігання фотографій картини до та після її реставрації. Фотографії зберігаються на файловому сервері, що дозволяє не лише переглядати їх у розробленій програмі, але і використовувати спеціалізовані сторонні програми, призначені для роботи з зображеннями.

Велика кількість звітів та інтуїтивно зрозумілий віконний інтерфейс роблять роботу користувачів швидкою та зручною. За необхідності в процесі експлуатації можливо додати нові звіти без необхідності міняти наявні.

					ДП ІС-5104.1260-с.ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

Можливі галузі застосування результатів проекту - картинні галереї України.

Дипломний проект присвячений розробці комплексу задач з обліку експонатів картинної галереї.

Практичне значення одержаних результатів. Розроблено програму обліку експонатів картинної галереї.

Публікації. Результати роботи не були опубліковані.

					ДП ІС-5104.1260-с.ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Опис предметного середовища

1.1.1 Опис процесу діяльності

Предметним середовищем є діяльність персоналу картинної галереї з обліку її експонатів. На рисунку 1.1 у вигляді діаграми станів показані всі основні стани системи, в яких вона чекає на відповідні дії персоналу.

В разі успішної авторизації користувач бачить головне меню, кнопки та таблицю зі списком експонатів. Перелік експонатів таблиці відображає лише ті з них, що відповідають поточним параметрам фільтрації.

Меню включає в себе всі можливі дії користувача. Кнопки повторюють функціонал найбільшу уживаних пунктів меню.

Якщо згідно ролі права користувача обмежені, деякі кнопки та пункти меню недоступні.

Дії користувача в системі можна розділити на дві великі частини: ті, що можуть виконуватись незалежно від наявності обраного рядка таблиці експонатів та ті, що можливі лише за наявності обраного рядка цієї таблиці.

До перших відносяться:

- створення нового експонату;
- формування звітів;
- вибір умов фільтрації;
- редагування довідників, в т.ч. довідника користувачів.

До других відносяться:

- редагування даних експонату;
- перегляд фотографій експонату;
- видалення експонату.

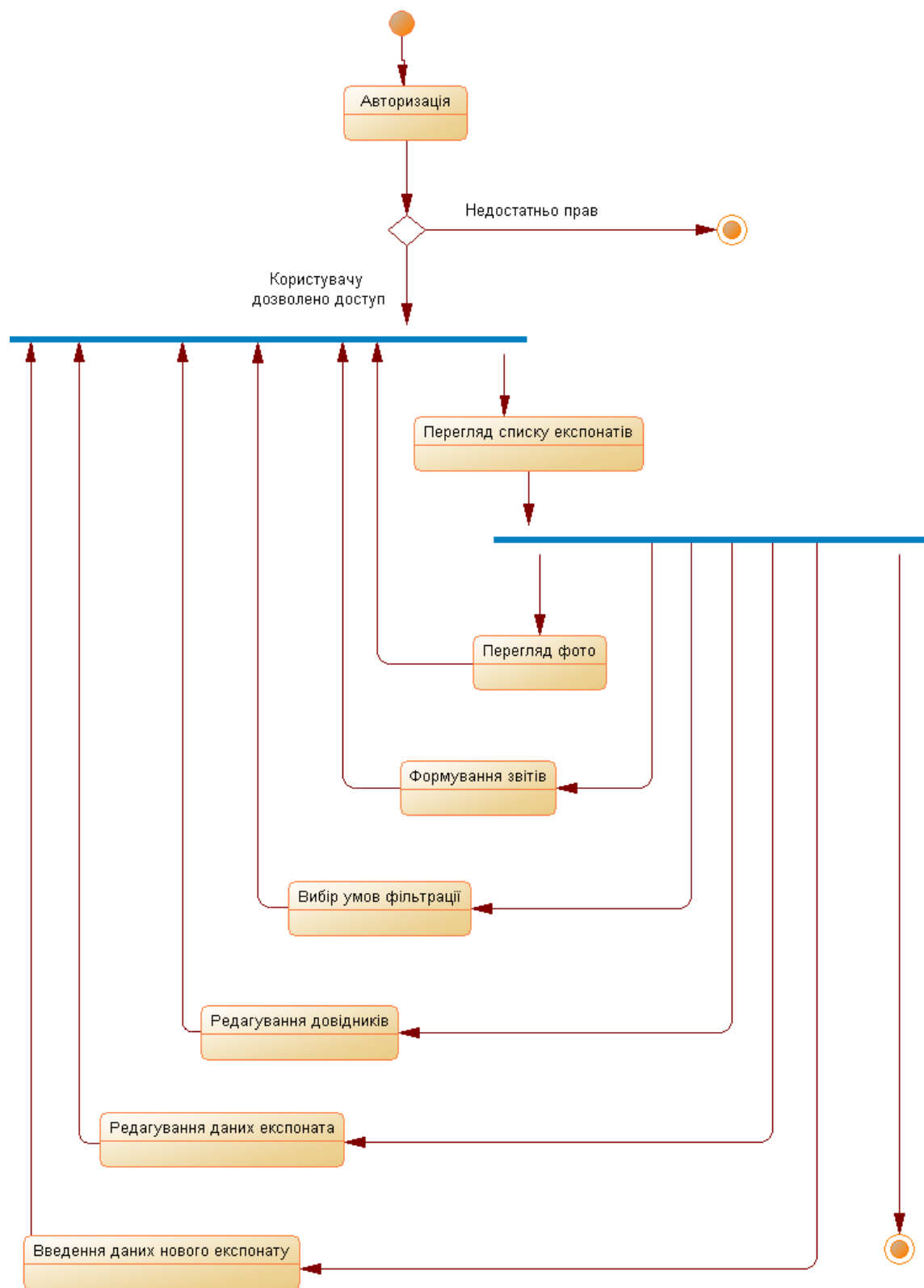


Рисунок 1.1 - Діаграма станів

Відповідна діаграма активності показана на рисунку 1.2, де показані основні дії користувача без обмеження його прав.

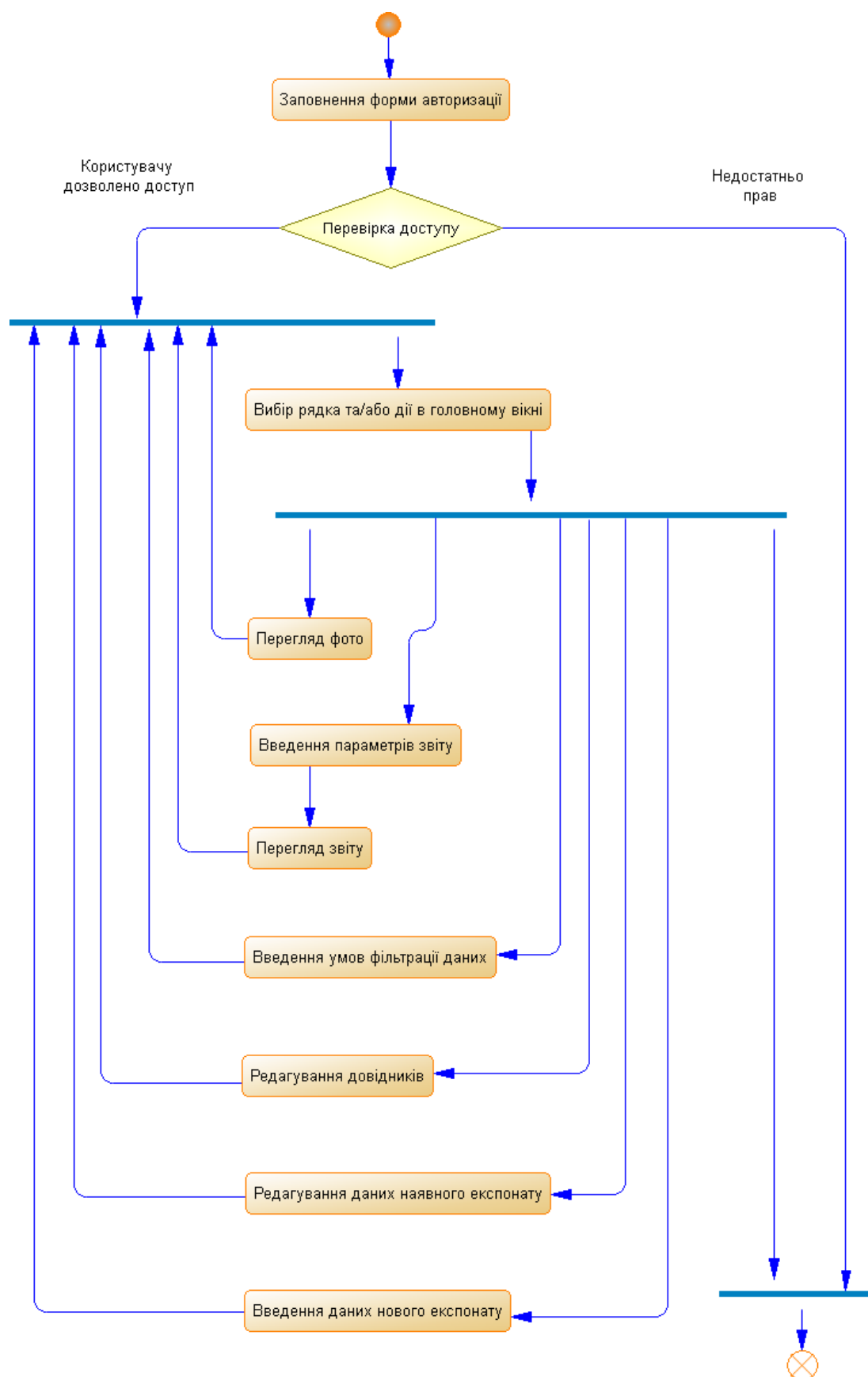


Рисунок 1.2 - Діаграма активності

1.1.2 Опис функціональної моделі

В межах розробленого програмного продукту будуть працювати актори (дійові особи) трьох ролей: адміністратор, менеджер, комірник. Окрема роль без доступу використовується, щоб заборонити доступ користувачу, не видаляючи його з довідника користувачів.

Модель варіантів використання побудована на рисунку 1.3. На найпоказані головні функції, які виконує кожний з акторів в програмі.

Адміністратор реєструє користувачів та, за необхідності, міняє їх доступ. Він має можливість редагувати довідники в разі відсутності менеджера. Технічно адміністратор має доступ і до іншого функціоналу програми, але в штатному режимі це не входить в його обов'язки, тому на рисунку 1.3 відповідні лінії відсутні.

Менеджер редагує довідники, формує звіти і може використовувати фільтр для пошуку експонатів, що відповідають певним критеріям. В цьому сенсі таблиця головного вікна програми для нього також виконує функції звіту з поточного розташування експонатів галереї.

Комірник виконує функції оператора з вводу та редагування всієї інформації, що відноситься до експонатів. На практиці зміна розташування експонату в програму вноситься після підписання відповідних документів матеріально відповідальними особами, тому в програму вноситься лише нове місце розташування, а ці особи не вказуються.

Таким чином, пріоритети функціональних вимог наступні: спочатку адміністратор реєструє користувачів, потім менеджер редагує дані довідників, потім кожний комірник вводить інформацію про експонати, за які несе відповідальність. На основі введенної інформації менеджер формує звіти.

Будь-який користувач може отримати ті дані, про обраний експонат, які на даний момент зберігаються в базі даних, навіть якщо інший користувач також працює з даними того самого експонату.

					ДП ІС-5104.1260-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

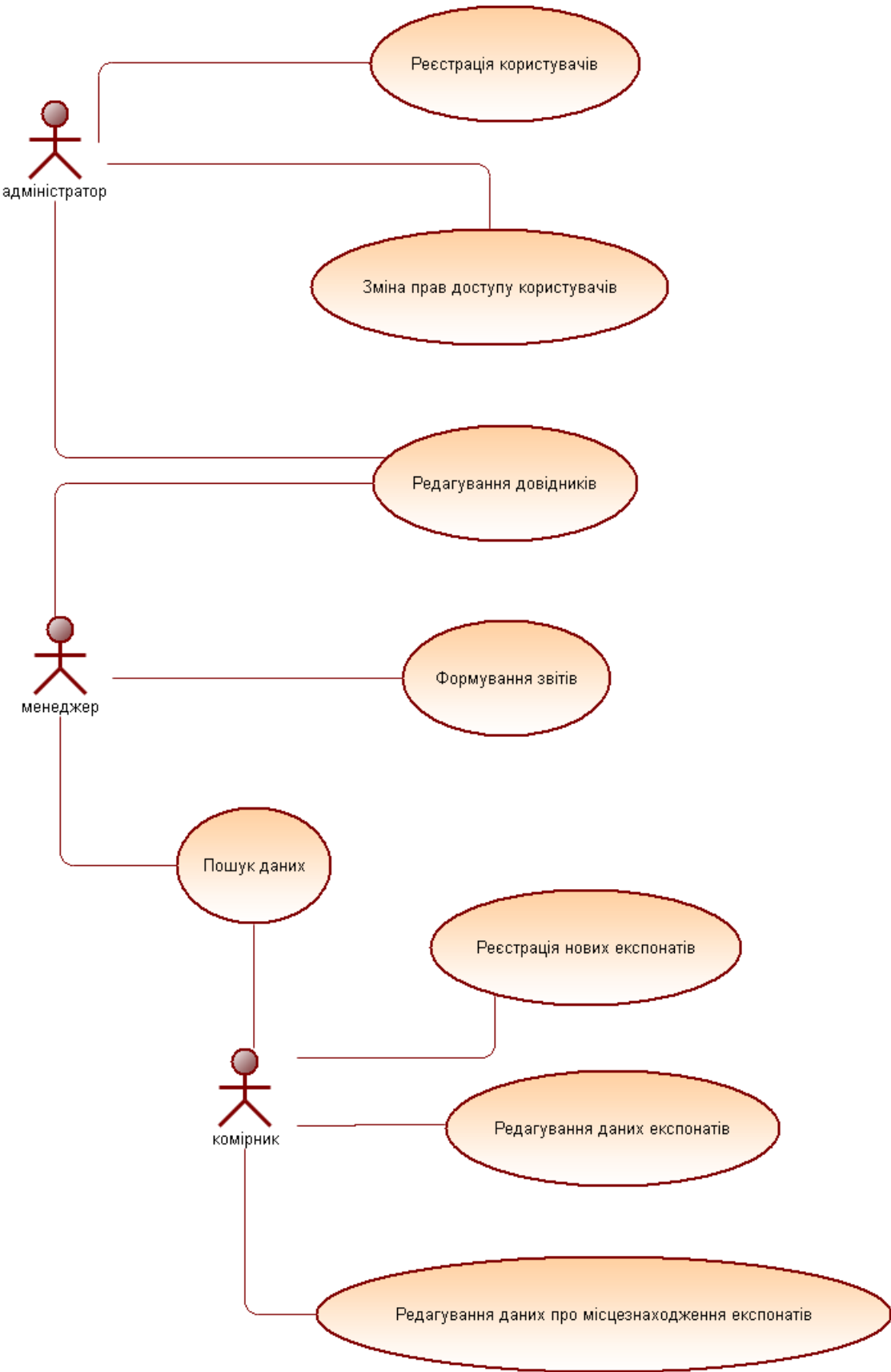


Рисунок 1.3 - Діаграма варіантів використання

У випадку спроби внести зміни одночасно кількома користувачами до одного експонату, конфлікт вирішується засобами СКБД. Зазвичай це означає, що зберігаються ті дані, які передані на сервер пізніше.

1.2 Огляд наявних аналогів

На ринку України наявна велика кількість програм складського обліку, що здатні частково автоматизувати облік експонатів картинних галерей, але відсутнє спеціалізоване програмне забезпечення, що враховує специфіку бізнес-процесів у цій сфері.

Наприклад, в [1] описана програма Система управління складом. Вона призначається в основному для торгових підприємств. Вона спрощує процес прийому, зберігання і реалізації з підготовкою відповідних документів. Також вона забезпечує контроль оплати від замовників і дозволяє проводити облік повернень. Програма формує необхідні товарні документи, накладні, договори, рахунки, акти і звіти.

Для підприємців [2] пропонує програму МойСклад, яка автоматизує процес обліку продаж, закупок, товарів та складу. Ця програма дає статистику продажів та оборотів і т.п. Тобто ця програма також орієнтована на торгівлю.

Згідно [3], професійна програма UniproRetail для обліку на складських об'єктах дозволяє вести повноцінний складський облік, включаючи такі особливості, як управління асортиментом та ціноутворенням, оптимізація складу, робота з постачальниками, формування звітів.

Одна зі спеціалізованих програм РемонтОнлайн представлена в [4], але вона орієнтована на ремонтні підприємства, що закупають та використовують великі переліки запчастин.

Велика програма УкрСклад, в [5] має перелік можливостей на кілька сторінок, але для картинної галереї більшість з них зайва, а інших, як то тимчасова передача експонату в оренду чи на реставрацію, в ній не вистачає.

					ДП ІС-5104.1260-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

Програма для складу з [6] пропонується малим та великим торговельним організаціям та може використовуватись мережевим торговельним бізнесом, що має філії у різних містах та країнах. На відміну від багатьох інших, в ній можна завантажити фотографії товару.

Складський калькулятор з [7], навпаки, призначена для персонального обліку. Програма зберігає код, найменування, ціну, дату останнього руху товару і залишки на складі, для кожного товару зберігаються всі його рухи.

Програма складського обліку Діловод з [8] призначена для складського обліку в сфері торгівлі, послуг та виробництва. Вона пропонує можливість налаштування інтерфейсу і рівня доступу до даних в залежності від ролей користувачів.

Таким чином, на ринку є багато програм складського обліку, які мають окремі функції, що були б зручні при обліку експонатів картинної галереї, але немає жодної, що відповідала б всім вимогам і мала всі ці функції.

1.3 Постановка задачі

1.3.1 Призначення розробки

Система призначена для автоматизації обліку експонатів картинної галереї, що не лише неодноразово переміщуються в її межах, але і тимчасово передаються або продаються назавжди іншим особам.

1.3.2 Цілі та задачі розробки

Розробка повинна забезпечувати такі результати:

- роботу кількох користувачів ЛОМ зі спільною базою даних;
- обмеження доступу користувачів згідно ролям;
- редагування довідників;
- редагування переліку експонатів;
- збереження історії переміщень експонатів;

- редагування переліку фотографій експонатів зі збереженням фотографій у файловій системі одного з файлових серверів ЛОМ;
- фільтрацію переліку експонатів за одним чи кількома одночасно параметрами;
- формування звітів (в експозиції, продано за період, вартість по приміщенням, кількість картин за країнами, кількість картин за авторами).

Для реалізації цих цілей треба вирішити такі задачі:

- визначити структуру програмного забезпечення;
- обрати технологію збереження даних;
- обрати вид інтерфейсу користувача;
- обрати технологію зберігання фотографій експонатів;
- сформуванати перелік важливих властивостей експонатів, які будуть вводитись в програму;
- сформуванати структуру звітів;
- обрати спосіб обмеження доступу користувачів;
- обрати програмні та технічні засоби реалізації проекту;
- створити програмний продукт;
- провести тестування створеного програмного продукту.

Висновок до розділу

В результаті роботи над цим розділом сформоване загальне уявлення про бізнес-процеси предметного середовища, що підлягають автоматизації. Проаналізована діяльність користувачів з різними обов'язками. Проведено огляд можливостей аналогічних програм, що представлені на ринку. Поставлені цілі та задачі, що мають бути вирішені для їх досягнення.

Цієї інформації достатньо для подальшої деталізації та використання в наступних розділах.

					ДП ІС-5104.1260-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Вхідні дані

Вхідні дані, що описують один експонат, представлені в таблиці 2.1. Деякі дані переносяться з рахунку-фактури (РФ), що оформляється при купівлі експонату, решту визначає комірник.

Таблиця 2.1

Вхідні дані експонату

Найменування	Тип даних	Розмірність	Міститься
інвентарний номер	текст	250	-
ПІБ автора	текст	150	РФ
країна автора	текст	50	РФ
примітка автора	текст	500	-
основа	текст	50	РФ
жанр	текст	50	-
техніка	текст	50	РФ
вид роботи	текст	50	РФ
поточне місцезнаходження	текст	50	-
набір фотографій	файли	-	-
ціна	дійсне число	20, 2	РФ
назва	текст	250	РФ
рік створення	ціле число	4	РФ
примітка експонату	текст	500	-
дата переміщення	дата	10	-
висота	ціле число	5	РФ
довжина	ціле число	5	РФ

Вхідні дані, що описують одного користувача, представлені в таблиці 2.2. Більшість даних вноситься на основі відповідних записок (ДЗ) начальників відповідних структурних підрозділів, затверджених керівником підприємства чи іншою уповноваженою особою. Логін та пароль адміністратор визначає самостійно.

Таблиця 2.2

Вхідні дані користувача

Найменування	Тип даних	Розмірність	Міститься
роль	текст	50	ДЗ
ПІБ	текст	150	ДЗ
логін	текст	50	-
пароль	текст	32	-
початок доступу	дата	10	ДЗ
кінець доступу	дата	10	ДЗ

2.2 Вихідні дані

Вихідні сигнали не використовуються. Вихідними документами програми є таблиці з переліком експонатів чи узагальненої інформації, сформованої на основі вибору користувача.

В цьому сенсі таблиця головного вікна програми з переліком експонатів також може розглядатися, як один з видів вихідних документів, що формується за обраними користувачем параметрами фільтрації. Перелік її реквізитів наведений в таблиці 2.3. Кожному експонату відповідає набір фотографій, перелік реквізитів яких наведений в таблиці 2.4.

Перелік експонатів та їх фотографії доступні всім користувачам, тому окремих вимог на доступність окремих реквізитів немає.

Таблиця 2.3

Вихідні реквізити переліку експонатів

Найменування	Тип даних	Розмірність
інвентарний номер	текст	250
назва	текст	250
рік створення	ціле число	4
ПІБ автора	текст	150
поточне місцезнаходження	текст	250
ціна, грн	дійсне число	20, 2
довжина, см	ціле число	5
висота, см	ціле число	5
жанр	текст	50
техніка	текст	50
вид роботи	текст	50
основа	текст	50
країна	текст	50
дата переміщення	дата	10
примітка експонату	текст	500

Таблиця 2.4

Вихідні реквізити переліку фотографій

Найменування	Тип даних	Розмірність
ім'я файлу	текст	250
примітка	текст	250
фотографія	графічне зображення	-

Перелік реквізитів звітів наведено в таблицях 2.5 - 2.9. Звіти в цілому доступні адміністратору та менеджеру, тому окремих вимог на доступність окремих реквізитів не мають.

Таблиця 2.5

Вихідні реквізити звіту "В експозиції"

Найменування	Тип даних	Розмірність
приміщення	текст	50
кількість картин	ціле число	5
кількість авторів	ціле число	5
кількість жанрів	ціле число	5
вартість, грн	дійсне число	20, 2

Таблиця 2.6

Вихідні реквізити звіту "Продано за період"

Найменування	Тип даних	Розмірність
автор	текст	150
країна	ціле число	5
кількість картин	ціле число	5
вартість, грн	дійсне число	20, 2

Таблиця 2.7

Вихідні реквізити звіту "Вартість по приміщенням"

Найменування	Тип даних	Розмірність
приміщення	текст	50
показ	текст	50
кількість картин	ціле число	5
вартість, грн	дійсне число	20, 2

Таблиця 2.8

Вихідні реквізити звіту "Кількість картин за країнами"

Найменування	Тип даних	Розмірність
країна	текст	50
приміщення	текст	50
кількість картин	ціле число	5
вартість, грн	дійсне число	20, 2

Таблиця 2.9

Вихідні реквізити звіту "Кількість картин за авторами"

Найменування	Тип даних	Розмірність
автор	текст	150
країна	текст	50
приміщення	текст	50
кількість картин	ціле число	5
вартість, грн	дійсне число	20, 2

2.3 Опис структури бази даних

Згідно [9 - 14] в предметній області виділено такі сутності: роль, користувач, країна, автор, картина, жанр, основа, рух, фотографія, техніка, вид роботи, місцезнаходження.

Ключові сутності та зв'язки між ними після приведення до третьої нормальної форми показані на рис. 2.1 у вигляді схеми бази даних.

Особливо варто відзначити, що сутності роль (rol) та користувач (myuser) не мають зв'язків з іншими сутностями БД. Вони необхідні для обліку користувачів та обмеження їх доступу у клієнтській частині розробленого комплексу задач згідно ролям.

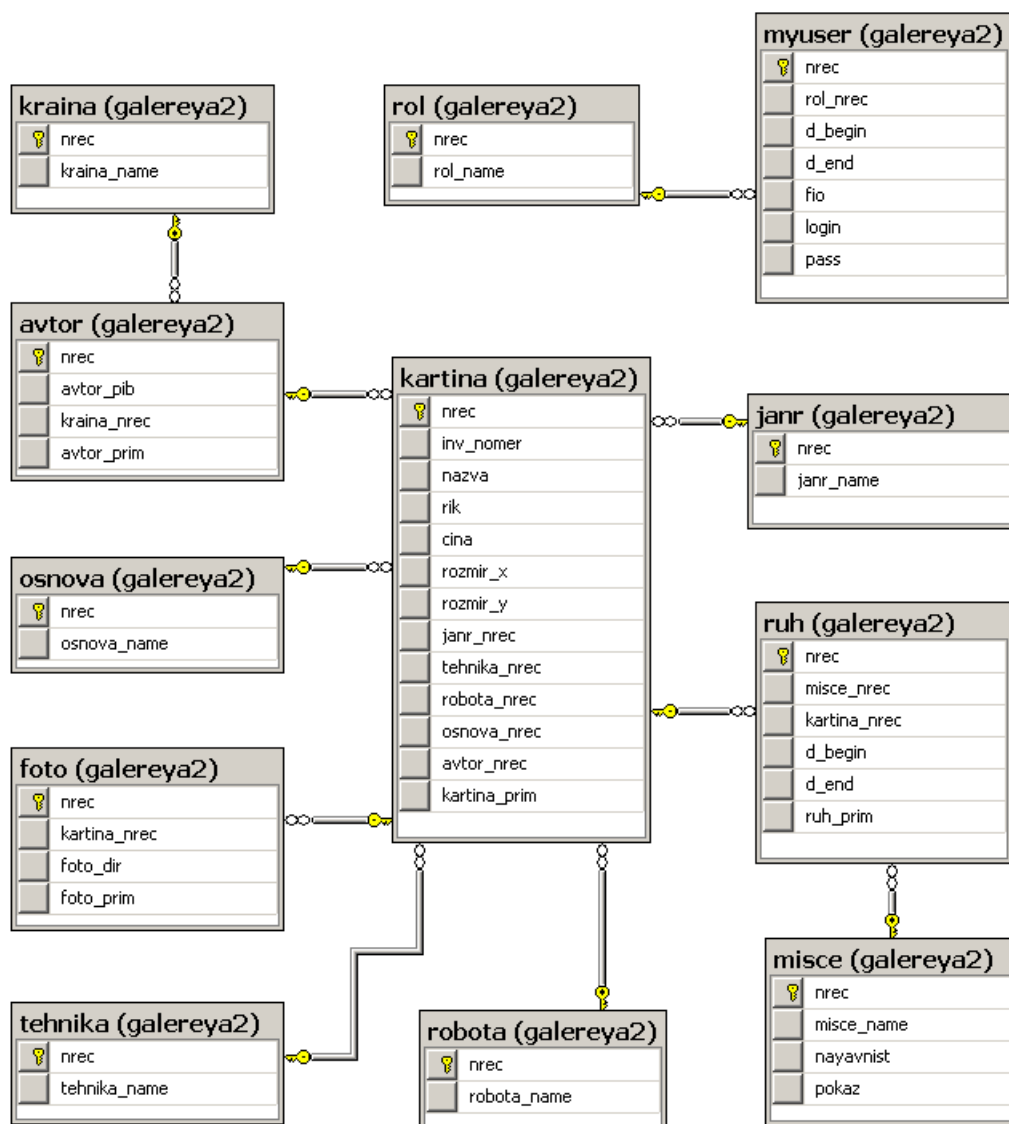


Рисунок 2.1 – Схема бази даних

База даних складається з 12 таблиць, до яких входить інформація про всі сутності предметної області. Атрибути представлено в таблиці 2.10.

Таблиця 2.10

Атрибути сутностей предметної області

1. Місцезнаходження		
Найменування	Тип даних	Розмірність
приміщення	текст	50
наявність	дійсне число	1, 0
показ	ціле число	1, 0
2. Рух		
дата початку переміщення	дата	10
дата закінчення переміщення	дата	10
примітка	текст	500
3. Жанр		
назва жанру	текст	50
4. Робота		
вид роботи	текст	50
5. Країна		
назва країни	текст	50
6. Автор		
ПІБ	текст	150
примітка	текст	500
7. Основа		
вид основи	текст	50
8. Техніка		
вид техніки	текст	50
9. Фото		
ім'я файлу	текст	250

Продовження таблиці 2.10

примітка	текст	500
фотографія	графічне зображення	-
10. Експонат		
інвентарний номер	текст	250
назва	текст	250
рік створення	ціле число	4
ціна	дійсне число	20, 2
висота	ціле число	5
довжина	ціле число	5
примітка	текст	500
11. Користувач		
початок доступу	дата	10
кінець доступу	дата	10
ПІБ	текст	150
логін	текст	50
пароль	текст	32
12. Роль		
назва ролі	текст	50

Висновок до розділу

У цьому розділі визначено всю початкову інформацію, яка потрапляє до програми та всю звітну інформацію, яку програма видає в результаті роботи. На основі цієї інформації створено структуру БД у 3НФ, яка повністю відповідає розробленому комплексу задач.

3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Змістовна постановка задачі

Програма повинна формувати всі звіти, перелічені у розділі 2 і зберігати всю необхідну для цього інформацію.

3.2 Математична постановка задачі

Всі звіти повинні формуватись в табличній формі.

3.3 Обґрунтування методу розв'язання

Оскільки дані предметної області зберігаються в реляційній БД, логічним є використання запитів мовою SQL для отримання результатів у табличному вигляді.

3.4 Опис методів розв'язання

Запит для отримання прав доступу (ролі) користувача приймає параметри логін та пароль: `SELECT nrec, rol_nrec, fio FROM [galereya].[galereya2].[myuser] WHERE login=? and pass = galereya2.md5(?) and GETDATE() between d_begin and d_end`. При цьому пароль шифрується на стороні сервера одностороннім алгоритмом MD5, що зменшує можливість дискредитації паролів.

Для формування таблиці головного вікна згідно з дев'ятьма параметрами фільтру, ці параметри передаються в процедуру, яка заздалегідь створена в БД і повертає результат запиту: `{call galereya2.p_kartina (?, ?, ?, ?, ?, ?, ?, ?, ?)}`.

Звіт "В експозиції" формується запитом: `SELECT DISTINCT m.misce_name, COUNT(1) as kilkist, COUNT(DISTINCT k.avtor_nrec) as k_avtor, COUNT(DISTINCT k.janr_nrec) as k_janr, SUM(k.cina) as vartist FROM galereya2.misce m, galereya2.ruh h, galereya2.kartina k WHERE h.d_end`

					ДП ІС-5104.1260-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24

IS NULL and m.pokaz = 1 and h.misce_nrec=m.nrec and h.kartina_nrec=k.nrec
GROUP BY m.misce_name ORDER BY m.misce_name.

Звіт "Продано за період" має два параметри - дати початку та кінця періоду і формується запитом: SELECT DISTINCT a.avtor_pib, n.kraina_name, COUNT(1) as kilkist, SUM(k.cina) as vartist FROM galereya2.misce m, galereya2.ruh h, galereya2.kartina k, galereya2.avtor a, galereya2.kraina n WHERE h.d_end IS NULL and (h.d_begin BETWEEN CAST(? AS datetime) AND CAST(? AS datetime)) and m.nrec = 10 and h.misce_nrec=m.nrec and h.kartina_nrec=k.nrec and k.avtor_nrec=a.nrec and a.kraina_nrec=n.nrec GROUP BY a.avtor_pib, n.kraina_name ORDER BY a.avtor_pib, n.kraina_name

Звіт "Вартість по приміщеннях" формується запитом: SELECT DISTINCT m.misce_name, CASE m.pokaz WHEN 1 THEN 'в експозиції' ELSE 'зберігається' END as pokaz_name, COUNT(1) as kilkist, SUM(k.cina) as vartist FROM galereya2.misce m, galereya2.ruh h, galereya2.kartina k WHERE h.d_end IS NULL and m.nayavnist = 1 and h.misce_nrec=m.nrec and h.kartina_nrec=k.nrec GROUP BY m.misce_name, m.pokaz ORDER BY m.misce_name.

Звіт "Кількість картин за країнами" формується запитом: SELECT DISTINCT n.kraina_name, m.misce_name, COUNT(1) as kilkist, SUM(k.cina) as vartist FROM galereya2.misce m, galereya2.ruh h, galereya2.kartina k, galereya2.avtor a, galereya2.kraina n WHERE h.d_end IS NULL and m.nayavnist = 1 and h.misce_nrec=m.nrec and h.kartina_nrec=k.nrec and k.avtor_nrec=a.nrec and a.kraina_nrec=n.nrec GROUP BY n.kraina_name, m.misce_name ORDER BY n.kraina_name, m.misce_name.

Звіт "Кількість картин за авторами" формується запитом: SELECT DISTINCT a.avtor_pib, n.kraina_name, m.misce_name, COUNT(1) as kilkist, SUM(k.cina) as vartist FROM galereya2.misce m, galereya2.ruh h, galereya2.kartina k, galereya2.avtor a, galereya2.kraina n WHERE h.d_end IS NULL and m.nayavnist = 1 and h.misce_nrec=m.nrec and h.kartina_nrec=k.nrec

					ДП ІС-5104.1260-с.ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

and k.avtor_nrec=a.nrec and a.kraina_nrec=n.nrec GROUP BY a.avtor_pib,
n.kraina_name, m.misce_name ORDER BY a.avtor_pib, n.kraina_name,
m.misce_name.

Всі отримані результати можуть відображатись в додатку користувача без додаткової математичної обробки.

Висновок до розділу

Створені в БД таблиці та процедури дозволяють отримати результати для відображення всіх необхідних звітів в табличній формі. Для формування запитів використовується мова SQL. Ці результати не потребують додаткової математичної обробки.

					ДП ІС-5104.1260-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

4.1 Засоби розробки

З огляду на клієнт-серверну архітектуру розробки, було обрано такі засоби розробки:

- сервер баз даних SQL Server 2008 Express;
- програма для візуального проектування БД Microsoft SQL Server Management Studio Express (SSMSE);
- програма для розробки клієнтського додатку з графічним інтерфейсом Microsoft Visual Studio 2010;
- мова програмування с#;
- підключення клієнта до сервера через драйвер ODBC.

4.2 Вимоги до технічного забезпечення

Сервер та клієнтські ПК повинні мати з'єднання, що забезпечує обмін даними зі швидкістю не менше 100 Мб/с.

Вимоги до серверу: оперативна пам'ять не менше 16Гб, процесор не менше 4 ядра по 2 ГГц, мережевий адаптер не менше 100 Мб/с, ОС Microsoft Windows Server 2008 (32- чи 64-розрядна) чи вище, СУБД Microsoft SQL Server 2008 Express чи вище.

Вимоги до клієнтського ПК: оперативна пам'ять не менше 1Гб, процесор не менше 1 ядра по 2 ГГц, мережевий адаптер не менше 100 Мб/с, ОС Microsoft Windows XP (32- чи 64-розрядна).

4.3 Архітектура програмного забезпечення

4.3.1 Діаграма класів

Діаграма класів показана на рис. 4.1. Оскільки розроблена програма має клієнт-серверну архітектуру, на стороні клієнта використовуються лише

					ДП ІС-5104.1260-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27

класи візуального інтерфейсу користувача, що відповідає різним видам діяльності користувача, описаній в п.1.1.1. Основні функції кожного класу наведено в табл.4.1.

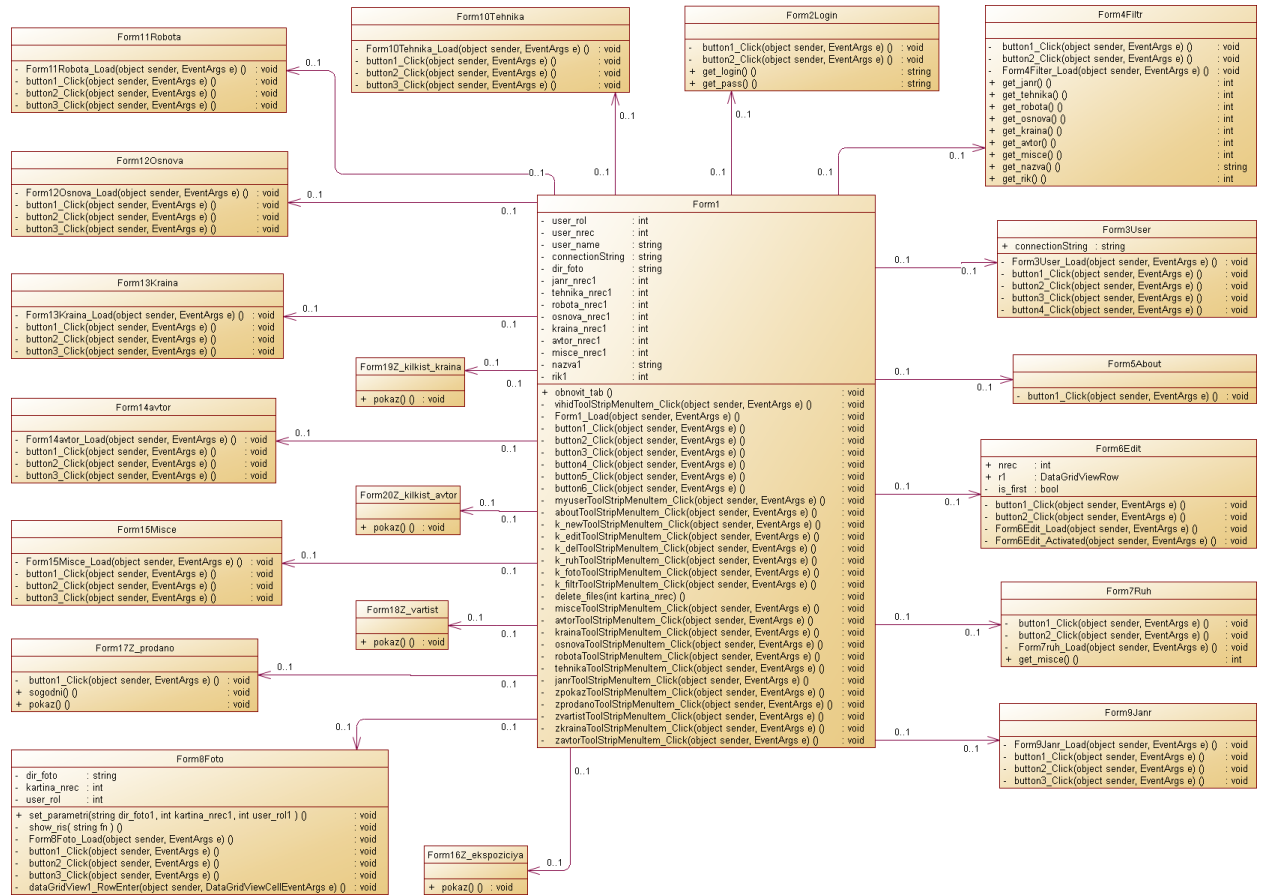


Рисунок 4.1 - Діаграма класів

Таблиця 4.1

Опис класів клієнтського додатку

Клас	Призначення
Form1	Головна форма, що дає доступ до решти функцій додатку через кнопки та меню
Form2Login	Авторизація користувача
Form3User	Редагування переліку користувачів
Form4Filtr	Вибір умов фільтру

Продовження таблиці 4.1

Form5About	Інформація про програму
Form6Edit	Редагування даних експонату
Form7Ruh	Переміщення експонату в інше місце
Form8Foto	Перегляд і редагування списку фотографій
Form9Janr	Редагування довідника жанрів
Form10Tehnika	Редагування довідника технік
Form11Robota	Редагування довідника виду робіт
Form12Osnova	Редагування довідника основ
Form13Kraina	Редагування довідника країн
Form14avtor	Редагування довідника авторів
Form15Misce	Редагування довідника місць знаходження
Form16Z_ekspoziciya	Формування звіту "В експозиції"
Form17Z_prodano	Формування звіту "Продано за період"
Form18Z_vartist	Формування звіту "Вартість по приміщенням"
Form19Z_kilkist_kraina	Формування звіту "Кількість картин за країнами"
Form20Z_kilkist_avtor	Формування звіту "Кількість картин за авторами"

4.3.2 Діаграма послідовності

Діаграма послідовності, що зображена на рис. 4.2, ілюструє взаємодію клієнтської та серверної частини комплексу задач. В узагальненій формі можна сказати, що клієнт посилає сервери запити кількох типів:

- при авторизації повертається один рядок даних з даними вказаного користувача або жодного рядка, якщо пара логін та пароль не знайдені. Це можна розглядати, як один з видів звітів, але в даному випадку користувач не бачить таблиці з результатами;

- при створенні рядка користувач отримує довідкові дані, наприклад, дані списків, що випадають, вносить дані, передає їх на сервер та отримує результат збереження;
- при редагуванні даних користувач отримує попередні дані, міняє їх, передає на сервер та отримує оновлений результат;
- при видаленні рядка користувач спочатку обирає його в отриманому переліку, посилає запит видалення на сервер та отримує результат після видалення;
- при формуванні звітів користувач передає на сервер параметри звіту та отримує результат, який може виявитись пустим.

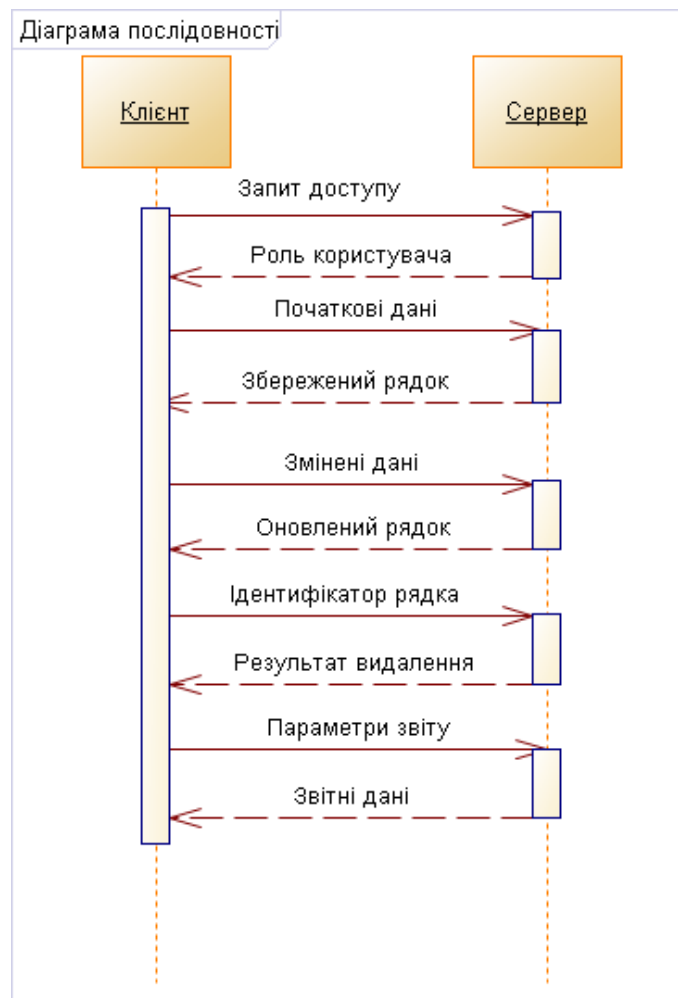


Рисунок 4.2 - Діаграма послідовності

4.3.3 Діаграма компонентів

На рис. 4.3 зображена діаграма компонентів комплексу задач. В укрупненому вигляді компонентів всього три: СКБД, драйвер ODBC та клієнтська програма.

При більш детальному розгляді в СКБД використовується БД galereya, а клієнтська програма складається з двадцяти класів, що забезпечують візуальний інтерфейс користувача.

Класи клієнтської програми на рис. 4.3 ті самі, що уже описані в пункті 4.3.1: Form1, Form2Login, Form3User, Form4Filtr, Form5About, Form6Edit, Form7Ruh, Form8Foto, Form9Janr, Form10Tehnika, Form11Robota, Form12Osnova, Form13Kraina, Form14avtor, Form15Miscе, Form16Z_ekspoziciya, Form17Z_prodano, Form18Z_vartist, Form19Z_kilkist_kraina, Form20Z_kilkist_avtor.

Клієнтська програма, в залежності від вибору користувача, використовує ті чи інші класи додатку. Можлива ситуація, коли деякі класи не використовуються взагалі. Наприклад, клас Form3User має право використовувати лише користувач, що має роль адміністратор.

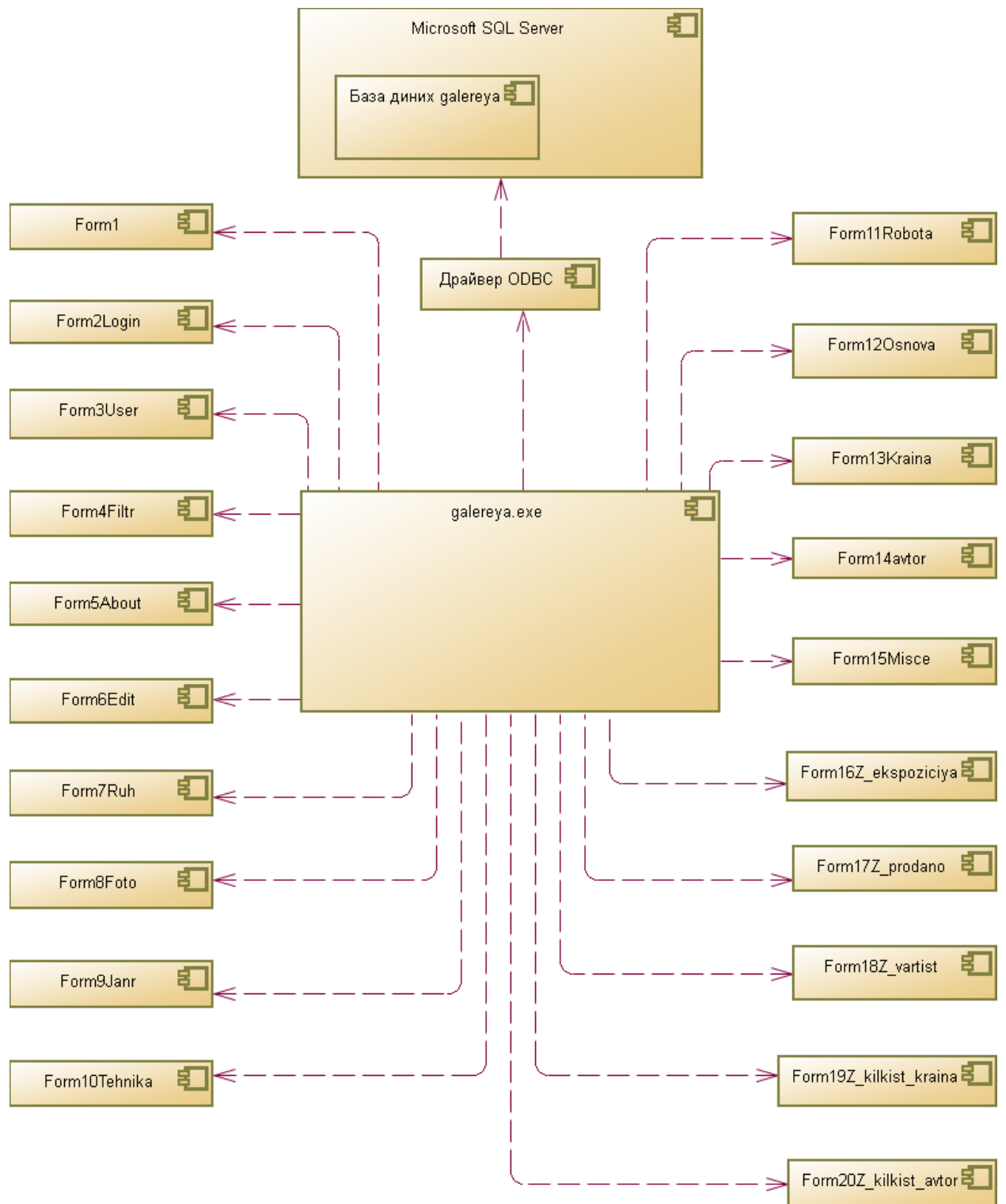


Рисунок 4.3 - Діаграма компонентів

4.3.4 Специфікація функцій

Функції класу Form1:

- public void obnovit_tab() //процедура формування даних про картини;
- private void vihidToolStripMenuItem_Click(object sender, EventArgs e)
//меню Вихід;
- private void Form1_Load(object sender, EventArgs e) //завантаження
цієї форми;
- private void button2_Click(object sender, EventArgs e) //кнопка
Редагувати;
- private void button1_Click(object sender, EventArgs e) //кнопка
Створити;
- private void button3_Click(object sender, EventArgs e) //кнопка
Видалити;
- private void button4_Click(object sender, EventArgs e) //кнопка
Перемістити;
- private void button5_Click(object sender, EventArgs e) //кнопка Фото;
- private void button6_Click(object sender, EventArgs e) //кнопка Фільтр;
- private void myuserToolStripMenuItem_Click(object sender, EventArgs
e) // меню Користувач;
- private void aboutToolStripMenuItem_Click(object sender, EventArgs e)
//меню Про програму;
- private void k_newToolStripMenuItem_Click(object sender, EventArgs
e) //меню Створити;
- private void k_editToolStripMenuItem_Click(object sender, EventArgs
e) //меню Редагувати;
- private void k_delToolStripMenuItem_Click(object sender, EventArgs e)
//меню Видалити;
- private void k_ruhToolStripMenuItem_Click(object sender, EventArgs e)
//меню Перемістити;

					ДП ІС-5104.1260-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

- private void k_fotoToolStripMenuItem_Click(object sender, EventArgs e) //меню Фото;
- private void k_filtrToolStripMenuItem_Click(object sender, EventArgs e) //меню Фільтр;
- private void delete_files(int kartina_nrec) //видалення всіх файлів фото картини;
- private void misceToolStripMenuItem_Click(object sender, EventArgs e) //меню Місце;
- private void avtorToolStripMenuItem_Click(object sender, EventArgs e) //меню Автор;
- private void krainaToolStripMenuItem_Click(object sender, EventArgs e) //меню Країна;
- private void osnovaToolStripMenuItem_Click(object sender, EventArgs e) //меню Основа;
- private void robotaToolStripMenuItem_Click(object sender, EventArgs e) //меню Робота;
- private void tehnikToolStripMenuItem_Click(object sender, EventArgs e) //меню Техника;
- private void janrToolStripMenuItem_Click(object sender, EventArgs e) //меню Жанр;
- private void zpokazToolStripMenuItem_Click(object sender, EventArgs e) //меню В експозиції;
- private void zprodanoToolStripMenuItem_Click(object sender, EventArgs e) //меню Продано за період;
- private void zvirtistToolStripMenuItem_Click(object sender, EventArgs e) //меню Вартість по приміщенням;
- private void zkrainaToolStripMenuItem_Click(object sender, EventArgs e) //меню Кількість картин за країнами;

- private void zavtorToolStripMenuItem_Click(object sender, EventArgs e) //меню Кількість картин за авторами.

Функції класу Form2Login:

- private void button1_Click(object sender, EventArgs e) //кнопка Увійти;
- private void button2_Click(object sender, EventArgs e) //кнопка Відміна;
- public string get_login() //відкритий метод для доступу до поля вводу логіна;
- public string get_pass() //відкритий метод для доступу до поля вводу паролю.

Функції класу Form3User:

- private void Form3User_Load(object sender, EventArgs e) //завантаження цієї форми;
- private void button3_Click(object sender, EventArgs e) //Зберегти зміни;
- private void button4_Click(object sender, EventArgs e) //Змінити пароль;
- private void button1_Click(object sender, EventArgs e) //Створити користувача;
- private void button2_Click(object sender, EventArgs e) //видалити користувача.

Функції класу Form4Filtr:

- private void Form4Filter_Load(object sender, EventArgs e) //завантаження цієї форми;
- private void button1_Click(object sender, EventArgs e) //кнопка ОК;
- private void button2_Click(object sender, EventArgs e) //кнопка Відміна;
- public int get_janr() //метод отримання вибору користувача - жанр;

- public int get_tehnika() //метод отримання вибору користувача - техніка;
- public int get_robota() //метод отримання вибору користувача - робота;
- public int get_osnova() //метод отримання вибору користувача - основа;
- public int get_kraina() //метод отримання вибору користувача - країна;
- public int get_avtor() //метод отримання вибору користувача - автор;
- public int get_misce() //метод отримання вибору користувача - місце знаходження;
- public string get_nazva() //метод отримання вибору користувача - частина назви;
- public int get_rik() //метод отримання вибору користувача - рік.

Функція класу Form5About private void button1_Click(object sender, EventArgs e) //кнопка ОК.

Функції класу Form6Edit:

- private void Form6Edit_Load(object sender, EventArgs e) //завантаження цієї форми;
- private void button1_Click(object sender, EventArgs e) //кнопка ОК;
- private void button2_Click(object sender, EventArgs e) //кнопка Відміна;
- private void Form6Edit_Activated(object sender, EventArgs e) // активація форми.

Функції класу Form7Ruh:

- private void Form7ruh_Load(object sender, EventArgs e) //завантаження цієї форми;
- private void button1_Click(object sender, EventArgs e) //кнопка ОК;

- private void button2_Click(object sender, EventArgs e) //кнопка Відміна;
- public int get_misce() //метод отримання вибору користувача - місце знаходження.

Функції класу Form8Foto:

- private void Form8Foto_Load(object sender, EventArgs e) //завантаження цієї форми;
- public void set_parametri(string dir_foto1, int kartina_nrec1, int user_rol1) // початкова настройка форми;
- private void show_ris(string fn) //показ файла;
- private void button1_Click(object sender, EventArgs e) //створити;
- private void button2_Click(object sender, EventArgs e) //видалити;
- private void button3_Click(object sender, EventArgs e) //зберегти зміни;
- private void dataGridView1_RowEnter(object sender, DataGridViewCellEventArgs e) //змінився обраний рядок.

Функції класу Form9Janr:

- private void Form9Janr_Load(object sender, EventArgs e) //завантаження цієї форми;
- private void button1_Click(object sender, EventArgs e) //створити рядок;
- private void button2_Click(object sender, EventArgs e) //видалити рядок;
- private void button3_Click(object sender, EventArgs e) //зберегти зміни.

Функції класу Form10Tehnika:

- private void Form10Tehnika_Load(object sender, EventArgs e) //завантаження цієї форми;
- private void button1_Click(object sender, EventArgs e) //створити рядок;

- private void button2_Click(object sender, EventArgs e) //видалити рядок;
- private void button3_Click(object sender, EventArgs e) //зберегти зміни.

Функції класу Form11Robota:

- private void Form11Robota_Load(object sender, EventArgs e) //завантаження цієї форми;
- private void button1_Click(object sender, EventArgs e) //створити рядок;
- private void button2_Click(object sender, EventArgs e) //видалити рядок;
- private void button3_Click(object sender, EventArgs e) //зберегти зміни.

Функції класу Form12Osнова:

- private void Form12Osнова_Load(object sender, EventArgs e) //завантаження цієї форми;
- private void button1_Click(object sender, EventArgs e) //створити рядок;
- private void button2_Click(object sender, EventArgs e) //видалити рядок;
- private void button3_Click(object sender, EventArgs e) //зберегти зміни.

Функції класу Form13Kraina:

- private void Form13Kraina_Load(object sender, EventArgs e) //завантаження цієї форми;
- private void button1_Click(object sender, EventArgs e) //створити рядок;
- private void button2_Click(object sender, EventArgs e) //видалити рядок;
- private void button3_Click(object sender, EventArgs e) //зберегти зміни.

Функції класу Form14avtor:

- private void Form14avtor_Load(object sender, EventArgs e)
//завантаження цієї форми;
- private void button1_Click(object sender, EventArgs e) //створити рядок;
- private void button2_Click(object sender, EventArgs e) //видалити рядок;
- private void button3_Click(object sender, EventArgs e) //зберегти зміни.

Функції класу Form15Miscе:

- private void Form15Miscе_Load(object sender, EventArgs e)
//завантаження цієї форми;
- private void button1_Click(object sender, EventArgs e) //створити рядок;
- private void button2_Click(object sender, EventArgs e) //видалити рядок;
- private void button3_Click(object sender, EventArgs e) //зберегти зміни.

Функція класу Form16Z_ekspoziciya - public void pokaz() //формувати.

Функції класу Form17Z_prodano:

- public void сьогодні();
- public void pokaz() //формувати;
- private void button1_Click(object sender, EventArgs e) //кнопка формувати.

Функція класу Form18Z_vartist - public void pokaz() //формувати.

Функція класу Form19Z_kilkist_kraina - public void pokaz() //формувати.

Функція класу Form20Z_kilkist_avtor - public void pokaz() //формувати.

4.4 Опис звітів

Звіт "В експозиції" показує по кожному експозиційному залу кількість картин, авторів, жанрів та сумарну вартість експонатів, що знаходяться там на даний момент.

					ДП ІС-5104.1260-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		39

Звіт "Продано за період" показує кількість та вартість проданих картин по кожному автору, чиї картини було продано за вказаний період.

Звіт "Вартість по приміщенням" показує по кожному приміщенню кількість картин та сумарну вартість експонатів, що знаходяться там на даний момент.

Звіт "Кількість картин за країнами" показує по кожному приміщенню кількість картин та сумарну вартість експонатів, що знаходяться там на даний момент в розрізі країн.

Звіт "Кількість картин за авторами" показує по кожному приміщенню кількість картин та сумарну вартість експонатів, що знаходяться там на даний момент в розрізі авторів.

Висновок до розділу

В цьому розділі за допомогою діаграм UML та словесного опису розглянуто вимоги до технічного забезпечення, обрану клієнт-серверну архітектуру програмного забезпечення, специфікації функцій клієнтської програми та склад звітів.

					ДП ІС-5104.1260-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		40

5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

5.1 Керівництво користувача

Неавторизовані користувачі не можуть увійти в програму. При спробі входу в програму відкривається вікно авторизації, показане на рис. 5.1.

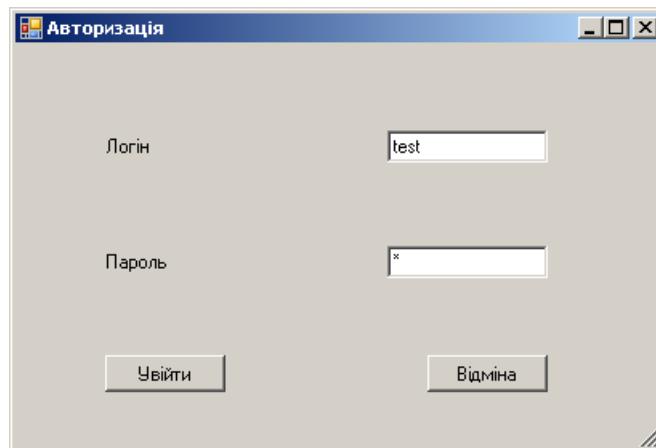


Рисунок 5.1 - Вікно авторизації

В разі успішної авторизації виводиться вікно, показане на рис. 5.2. Як видно, в ньому виводиться ПІБ користувача, отримане з бази даних, а не логін, введений в форму авторизації.

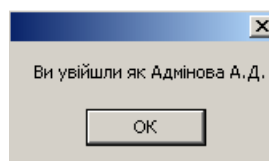


Рисунок 5.2 - Вікно повідомлення про вхід

Головне вікно програми показане на рис. 5.3 для користувача ролі адміністратор. У інших користувачів деякі кнопки та меню недоступні, згідно їхнім ролям.

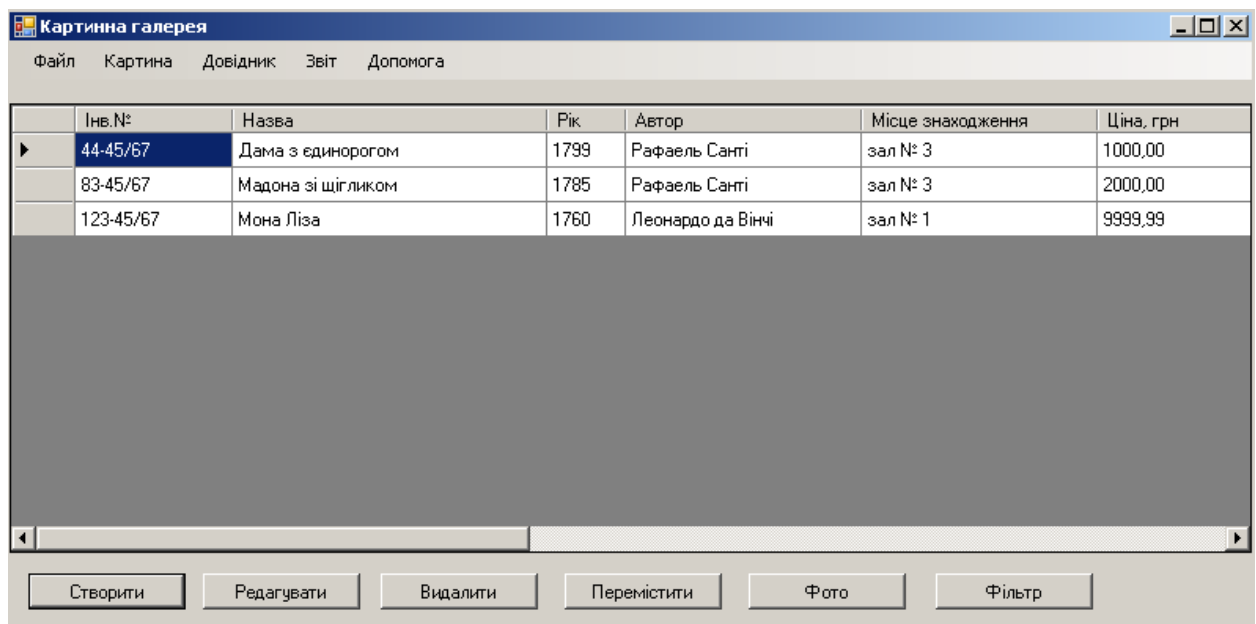


Рисунок 5.3 - Головне вікно програми

Вікно вводу даних нової картини показане на рис. 5.4 з прикладом введених даних. Місце розміщення і більшість інших полів заповнюються зі списків, що формуються з довідників. Це зменшує можливість помилки і прискорює роботу користувача.

Жанр: портрет

Техніка: пастель

Робота: оригінал

Основа: холст

Інв. №: 12-24/55

Ціна, грн: 70000

Місце розміщення: склад № 2

Назва: Автопортрет

Автор: Шевченко Т.Г.

Рік: 1860

Довжина, см: 1000

Висота, см: 13500

Примітка:

OK Відміна

Рисунок 5.4 - Вікно вводу даних нової картини

При редагуванні даних експонату бачимо на рис. 5.5 майже таке ж вікно, як при вводі нового, але в ньому дані про розміщення відсутні. Зміна розміщення описана окремо нижче.

The 'Картина' window contains the following fields:

- Жанр: портрет
- Техніка: пастель
- Робота: оригінал
- Основа: холст
- Інв. №: 12-24/55
- Ціна, грн: 70000,00
- Назва: Автопортрет
- Автор: Шевченко Т.Г.
- Рік: 1860
- Довжина, см: 48
- Висота, см: 59
- Примітка: в овалі

Buttons: OK, Відміна

Рисунок 5.5 - Вікно редагування даних картини

Вікно перегляду фотографій, зображене на рис. 5.6, також дозволяє додавати чи видаляти фотографії. При додаванні відкривається стандартний діалог відкриття файлу, показаний на рис. 5.7.

The 'Фото' window displays a list of files in a table:

Файл	Примітка
2.jpg	
ml11.jpg	

Buttons: Створити, Видалити, Зберегти зміни

Preview image: Mona Lisa

Рисунок 5.6 - Вікно перегляду фотографій

Змн.	Арк.	№ докум.	Підпис	Дата

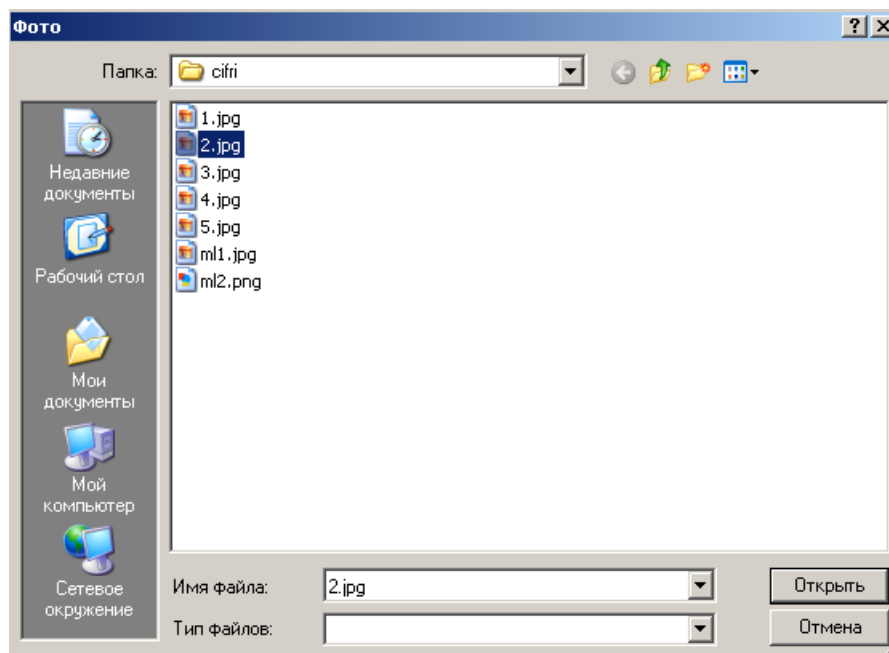


Рисунок 5.7 - Вікно вибору файлу фотографії

У випадку спроби видалення, коли не обрано жодного рядка таблиці, виводиться повідомлення, зображена на рис. 5.8 і програма продовжує роботу.

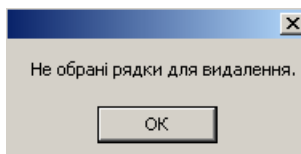


Рисунок 5.8 - Вікно повідомлення про неможливість видалення

Також файли фотографій видаляються автоматично при видаленні експонату, тому видалення картини в головному вікні програми відбувається у два кроки, як показано на рис. 5.9 та 5.10 - спочатку виводяться повідомлення про видалення файлів, а потім - самої картини. Після цього таблиця оновлюється автоматично.

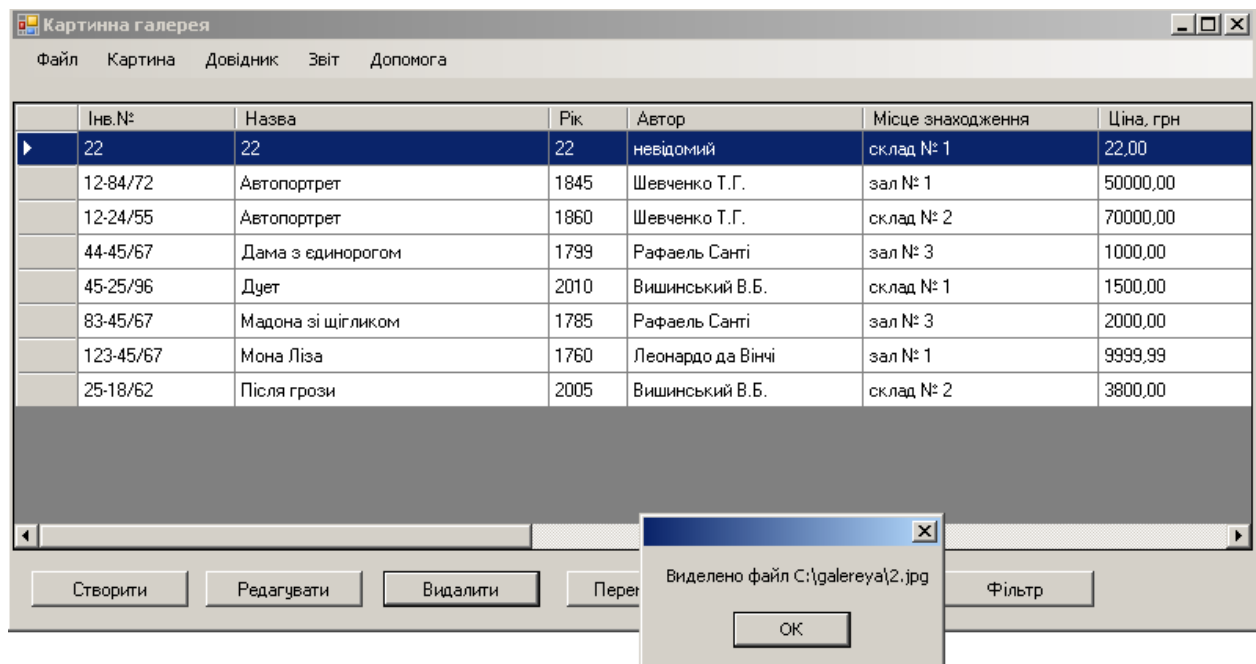


Рисунок 5.9 - Видалення виділеної картини - крок перший

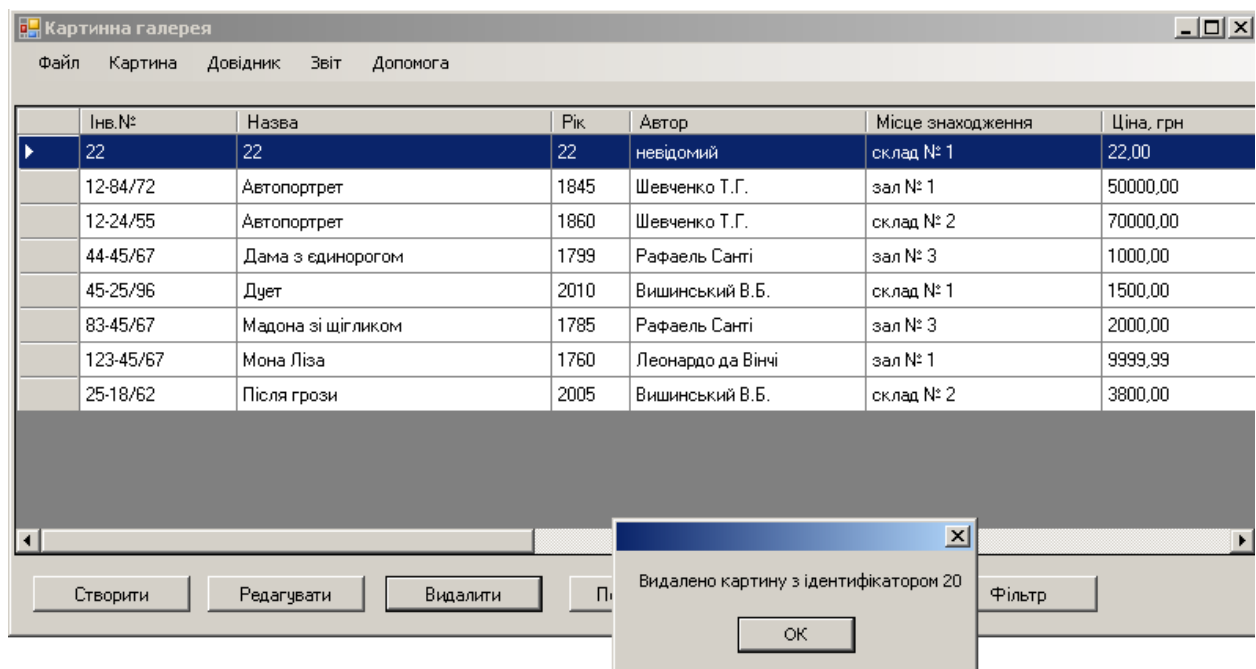


Рисунок 5.10 - Видалення виділеної картини - крок другий

Переміщення експоната вимагає збереження історії і є доволі частою операцією, тому винесене в окреме вікно, показане на рис. 5.11.

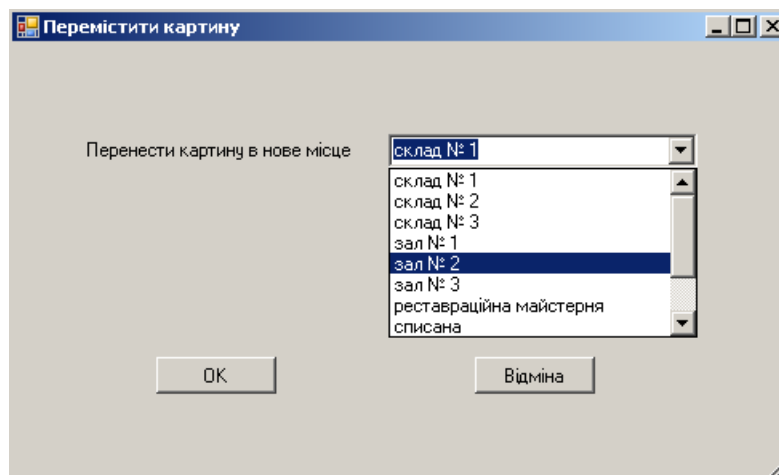


Рисунок 5.11 - Вікно переміщення картин

Для перегляду всіх експонатів, що відповідають одночасно кільком критеріям, в програмі передбачено багато умов фільтрації. Наприклад, на рис. 5.12 обрані умови для показу всіх портретів автора Шевченко Т.Г. Результат фільтрації буде показано в головному вікні програми, як на рис. 5.13.

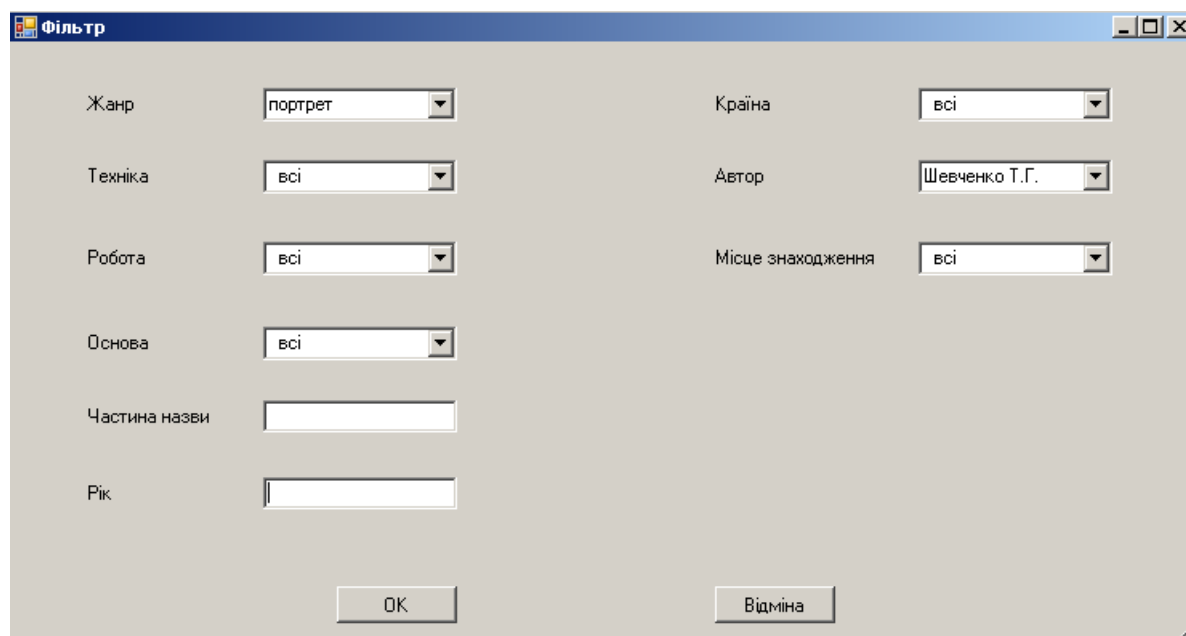


Рисунок 5.12 - Вікно вибору умов фільтрації даних з умовою портрети Шевченка

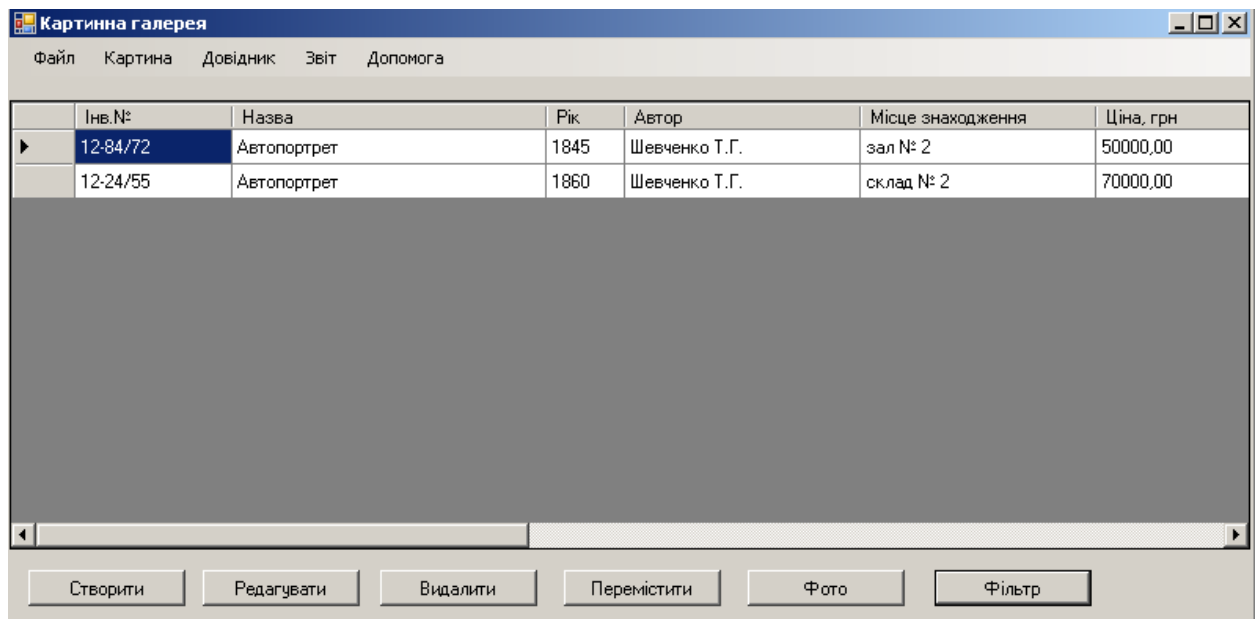


Рисунок 5.13 - Головне вікно з результатом фільтрації за умовою портрети Шевченка

Для перегляду всіх експонатів достатньо обрати умови фільтру, як на рис. 5.14.

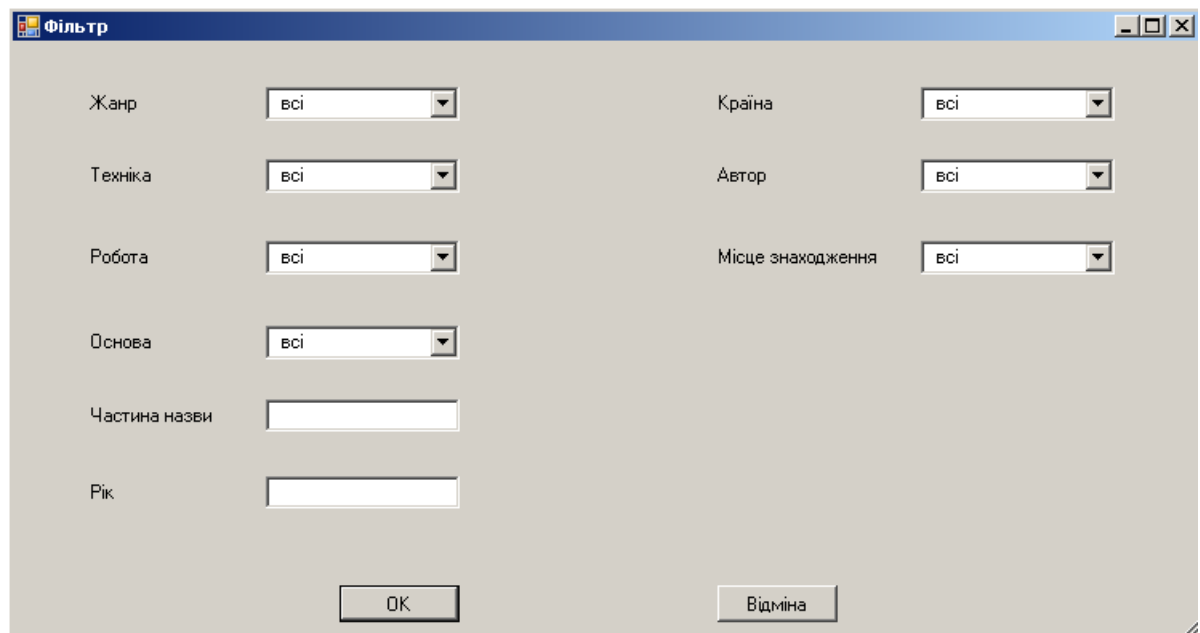


Рисунок 5.14 - Вікно вибору умов фільтрації даних з умовою вибору всіх картин

Програма має меню, що забезпечує доступ до всіх її функцій, якщо це дозволено роллю користувача. Меню Файл, показане на рис. 5.15, має єдину команду Вихід - для завершення роботи програми.

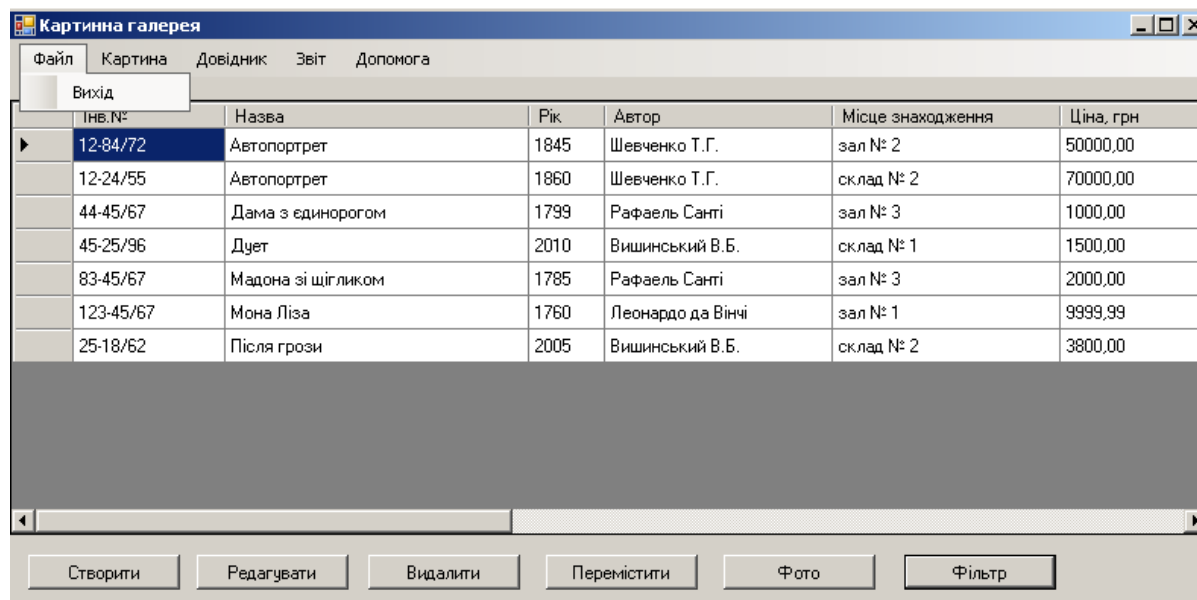


Рисунок 5.15 - Меню Файл

Меню Картина, показане на рис. 5.16, має всі команди для роботи з експонатом і вони ж доступні через кнопки внизу головного вікна.

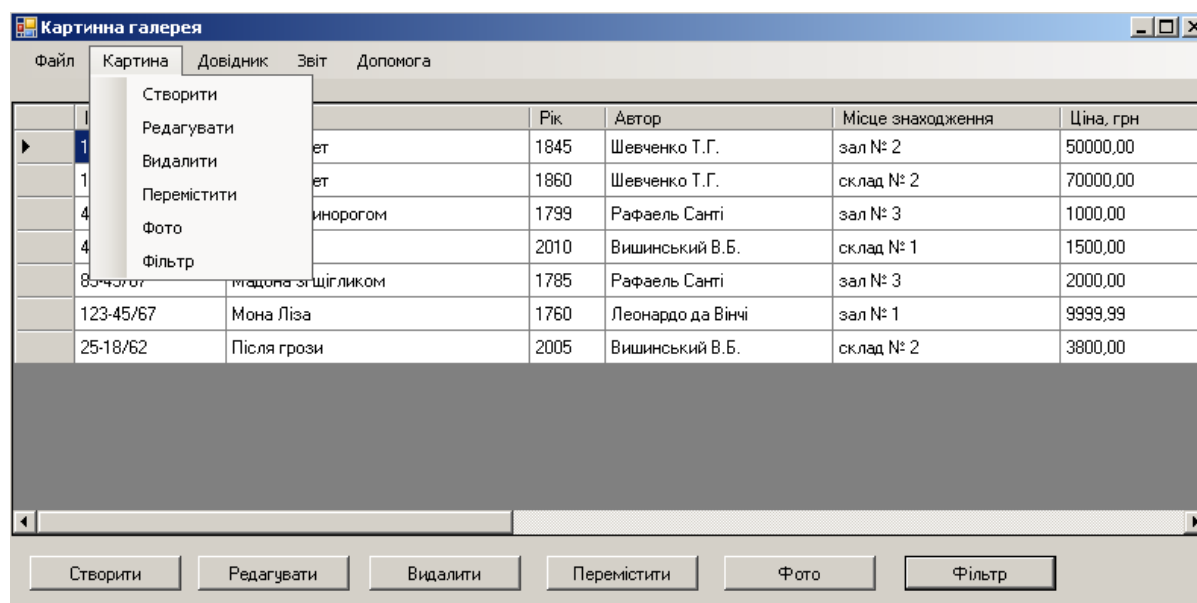


Рисунок 5.16 - Меню Картина

Меню Довідник, зображене на рис. 5.17, дозволяє редагувати всі довідники. При цьому довідник користувачів доступний лише адміністратору, решта - адміністратору і менеджеру.

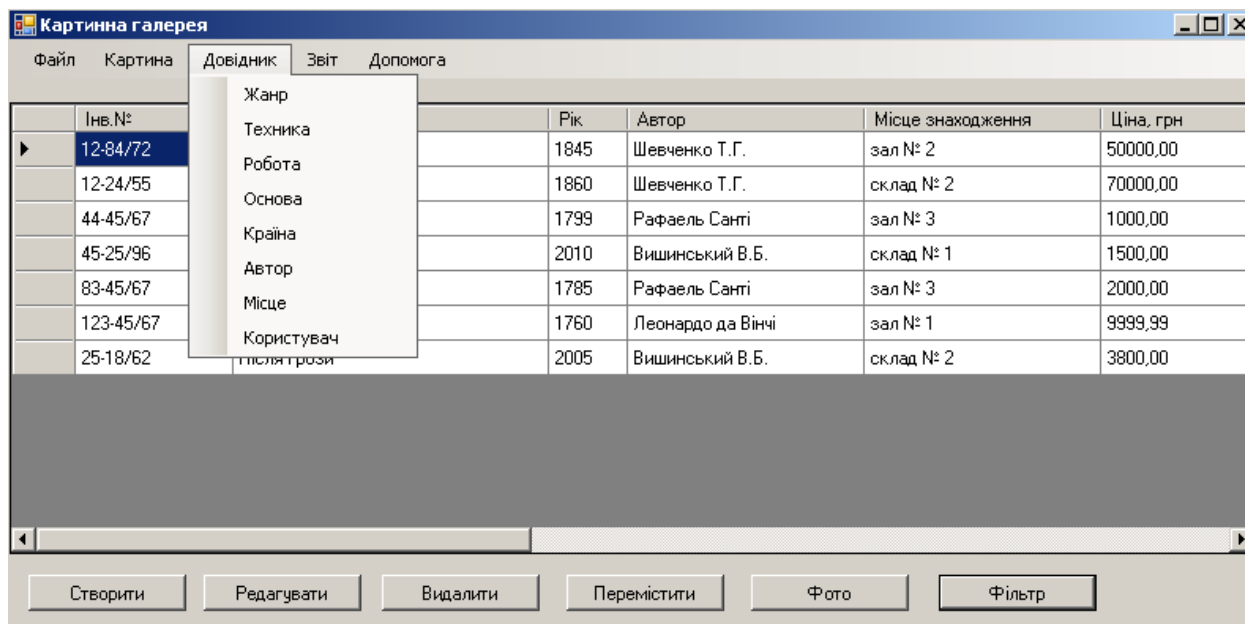


Рисунок 5.17 - Меню Довідник

Меню Звіт, показане на рис. 5.18, доступне адміністратору і менеджеру і служить в основному для отримання узагальненої інформації.

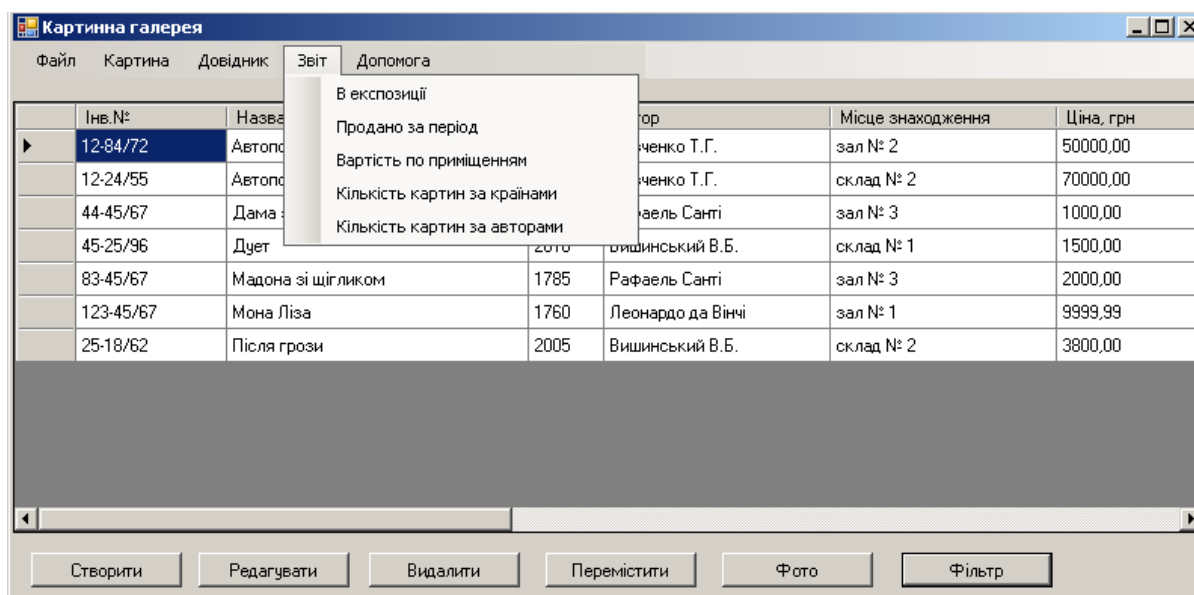


Рисунок 5.18 - Меню Звіт

Меню Допомога, показане на рис. 5.19, дає інформацію про програму.

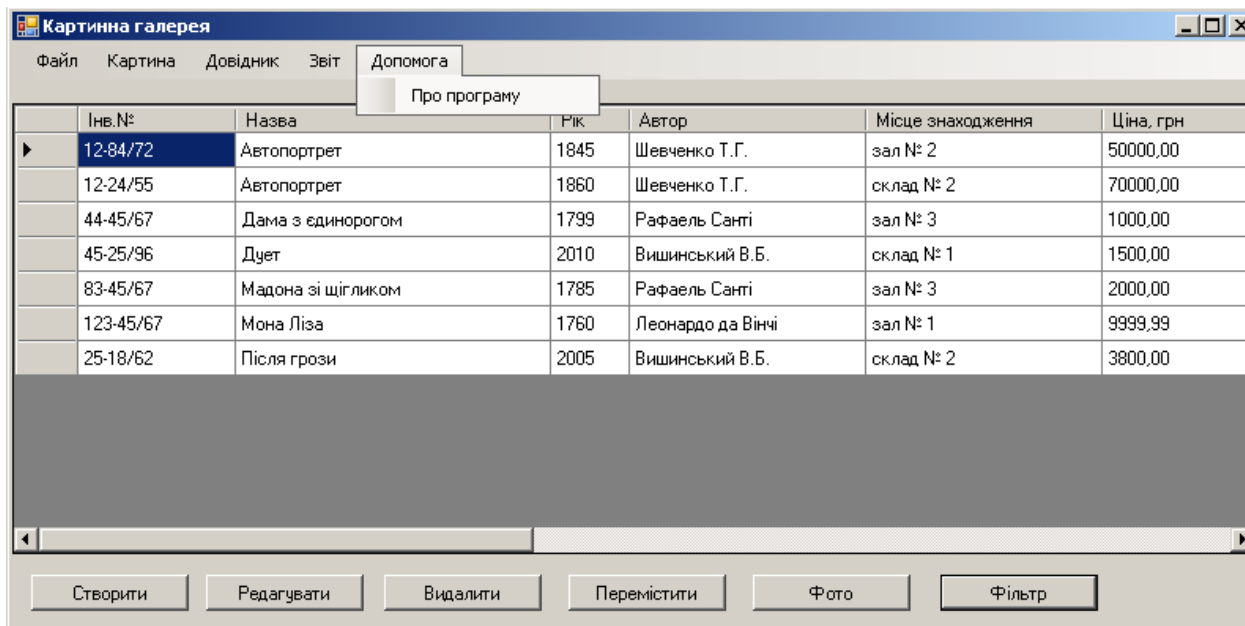


Рисунок 5.19 - Меню Допомога

Програма дозволяє налаштувати довідники відповідно до потреб конкретної картинної галереї.

Вікно редагування довідника жанрів показане на рис. 5.20.

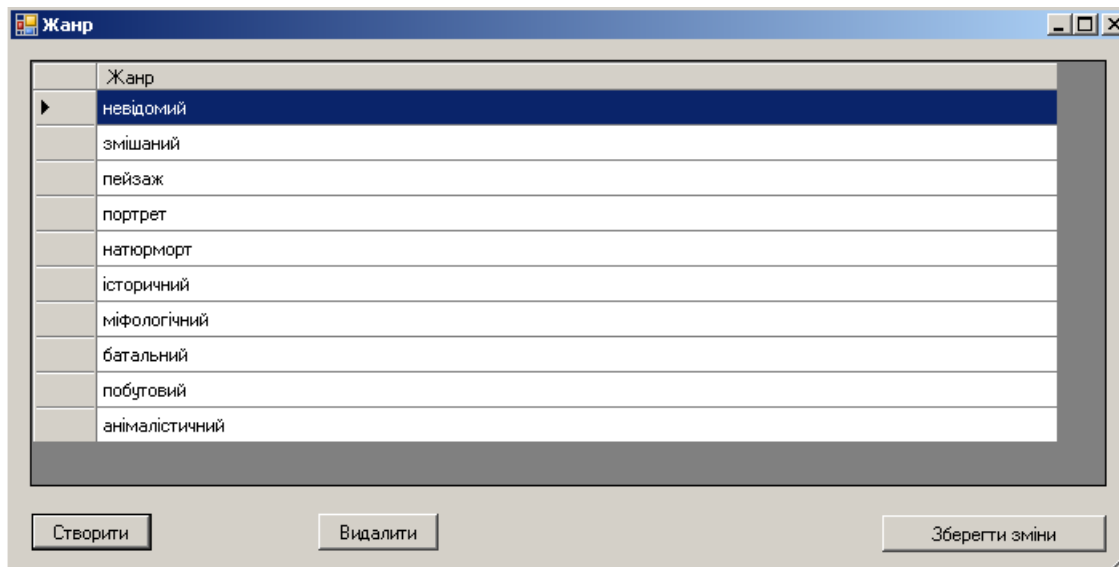


Рисунок 5.20 - Вікно довідника жанрів

Вікно редагування довідника технік показано на рис. 5.21.

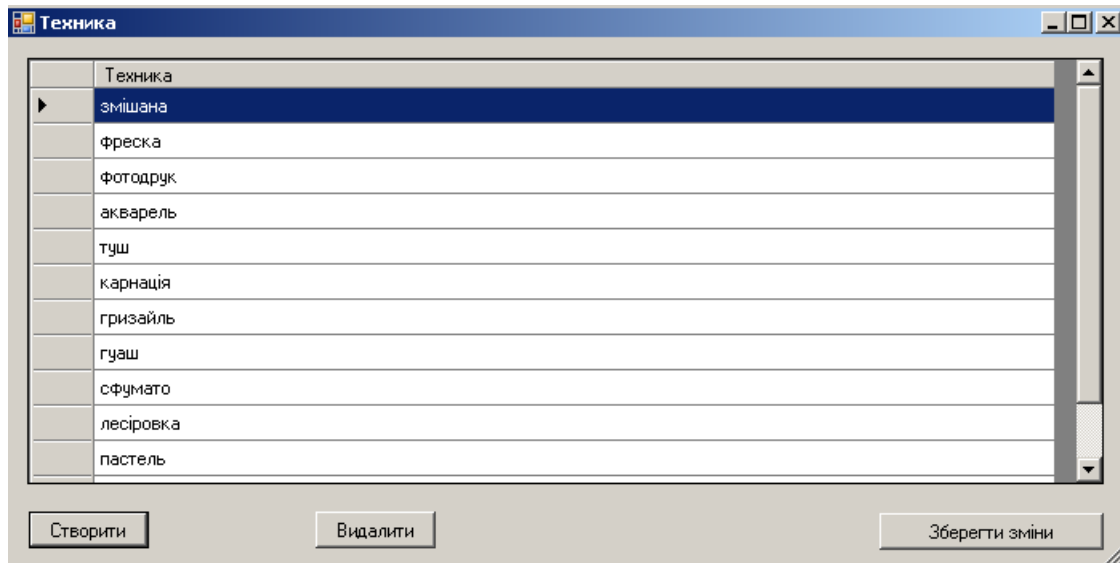


Рисунок 5.21 - Вікно довідника техніки малювання

Вікно редагування довідника видів робіт показано на рис. 5.22.

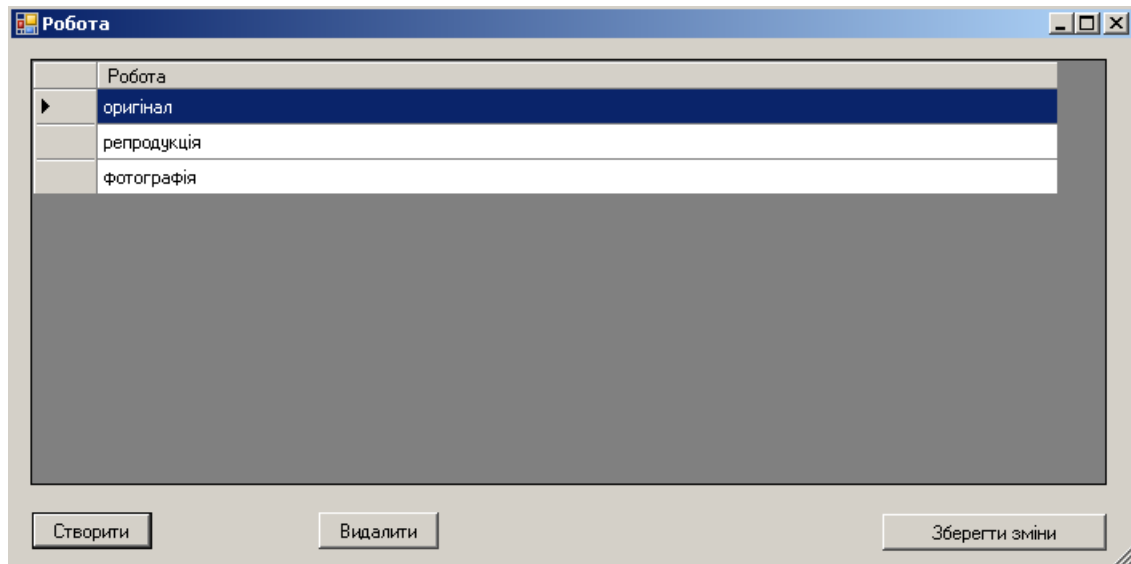


Рисунок 5.22 - Вікно довідника виду роботи

Вікно редагування довідника основ показано на рис. 5.23.

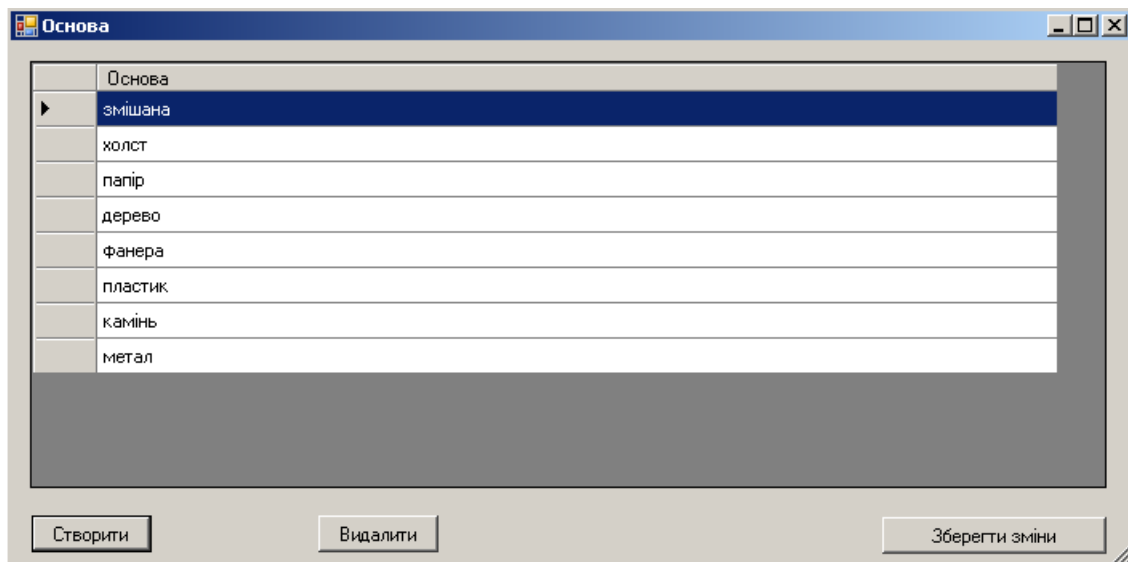


Рисунок 5.23 - Вікно довідника основ

Вікно редагування довідника країн показане на рис. 5.24.

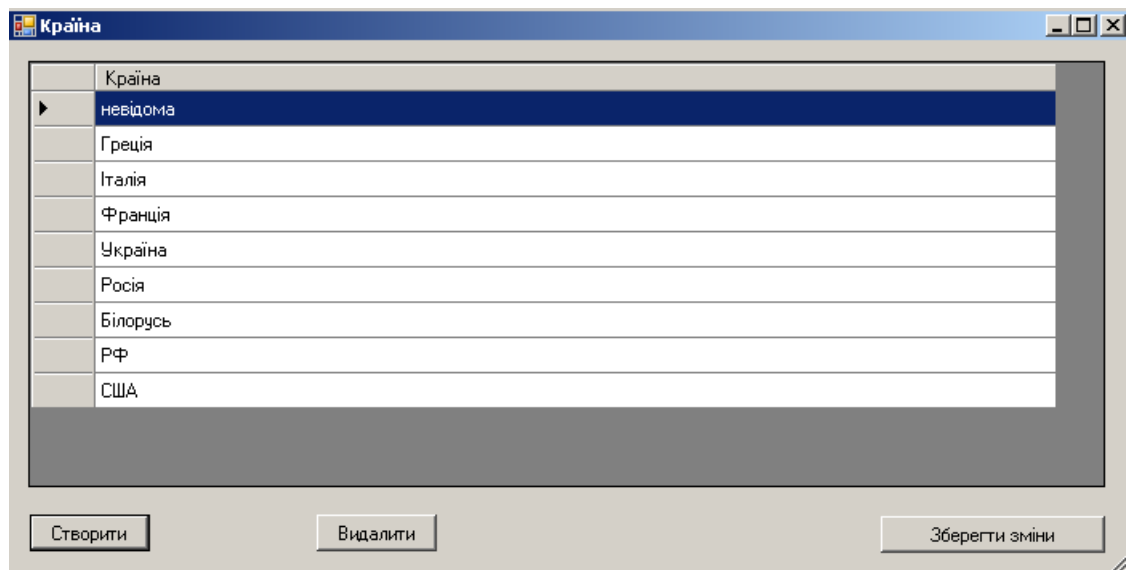


Рисунок 5.24 - Вікно довідника країн

Вікно редагування довідника авторів показане на рис. 5.25. В ньому країна береться з довідника країн.

ПІБ	Країна	Примітка
невідомий	невідомо	невідомий
Мікеланджело Буонароті	Італія	Мікеланджело ді Франческо ді Нері ді Мініато дель Сер...
Леонардо да Вінчі	Італія	15 квітня 1452, селище Анк'яно, поблизу Флоренції – 2 ...
Рафаель Санті	Італія	28 березня чи 6 квітня 1483, Урбіно – 6 квітня 1520, Рим
Шевченко Т.Г.	Україна	
Вишинський В.Б.	Росія	
Петренко С.Д.	Україна	

Створити Видалити Зберегти зміни

Рисунок 5.25 - Вікно довідника авторів

Вікно редагування довідника місць розташування показане на рис. 5.25. В тому числі в ньому передбачені можливості обліку продажу, тимчасової передачі чи викрадення експонатів.

Місце	Наявність	В показі
склад № 1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
склад № 2	<input checked="" type="checkbox"/>	<input type="checkbox"/>
склад № 3	<input checked="" type="checkbox"/>	<input type="checkbox"/>
зал № 1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
зал № 2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
зал № 3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
реставраційна майстерня	<input checked="" type="checkbox"/>	<input type="checkbox"/>
списана	<input type="checkbox"/>	<input type="checkbox"/>
викрадена	<input type="checkbox"/>	<input type="checkbox"/>
продана	<input type="checkbox"/>	<input type="checkbox"/>
тимчасово передана	<input type="checkbox"/>	<input type="checkbox"/>

Створити Видалити Зберегти зміни

Рисунок 5.26 - Вікно довідника місць розташування

Вікно редагування довідника користувачів показано на рис. 5.27. В ньому роль обирається зі списку. Список ролей користувач не може редагувати і не може змінити права кожної ролі.

The screenshot shows a window titled 'Користувач' (User). It contains a table with the following data:

Роль	Початок	Кінець	ПІБ	Логін
адміністратор	01.01.2019	31.12.2199	Адмінова А.Д.	1
менеджер	01.01.2019	31.12.2199	Менеджеров М.Е.	2
комірник	01.01.2019	31.12.2199	Комірник К.О.	3
перегляд	01.01.2019	31.12.2199	Переглядова П.Е.	4

Below the table, there are input fields for 'Пароль' (Password) and 'Пароль повторно' (Repeat password), and buttons for 'Зберегти пароль' (Save password), 'Створити' (Create), 'Видалити' (Delete), and 'Зберегти зміни' (Save changes).

Рисунок 5.27 - Вікно довідника користувачів

Оскільки пароль зберігається в базі даних в шифрованому вигляді, користувач не може його побачити, може лише змінити на новий. Для цього внизу вікна є відповідні поля. Пароль після збереження неможливо взяти, тому програма вимагає повторного вводу паролю.

Якщо пароль і його повтор не співпадають, збереження не відбувається. Натомість виводиться повідомлення, як показано на рис. 2.28.

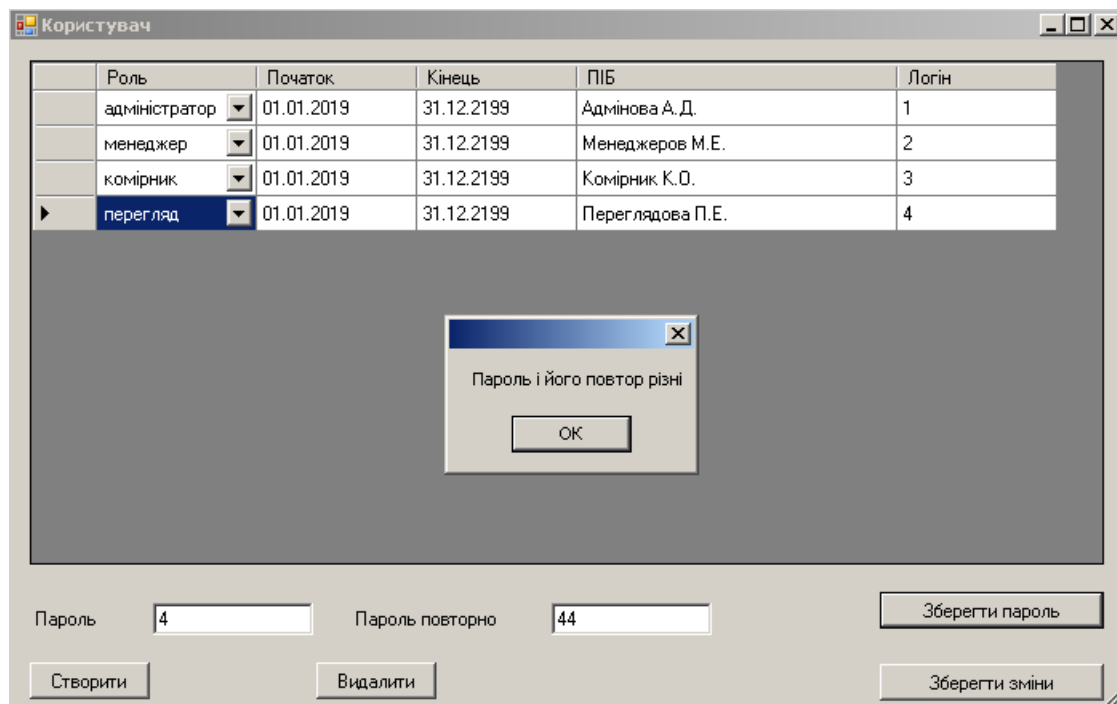


Рисунок 5.28 - Повідомлення про помилку в паролі

При успішній зміні паролю також виводиться повідомлення, як показано на рис. 2.29.

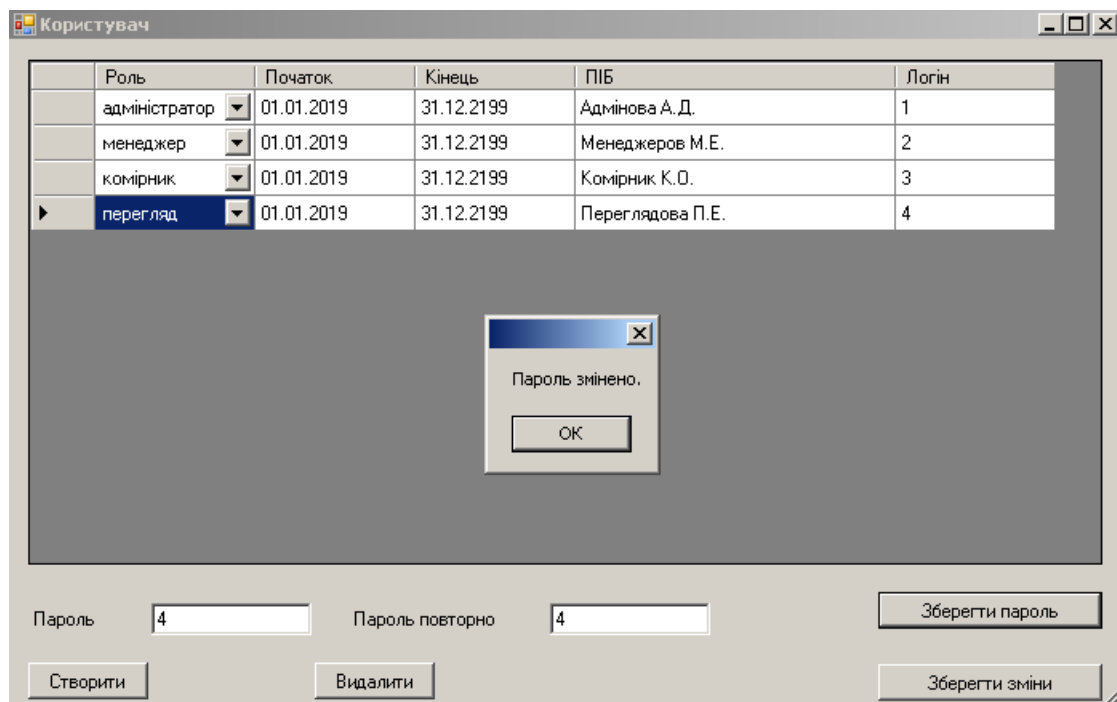


Рисунок 5.29 - Повідомлення про успішну зміну паролю

Всі звіти дають сумарну інформацію. Докладно список експонатів кожного залу галереї можна побачити в головному вікні програми за допомогою фільтру.

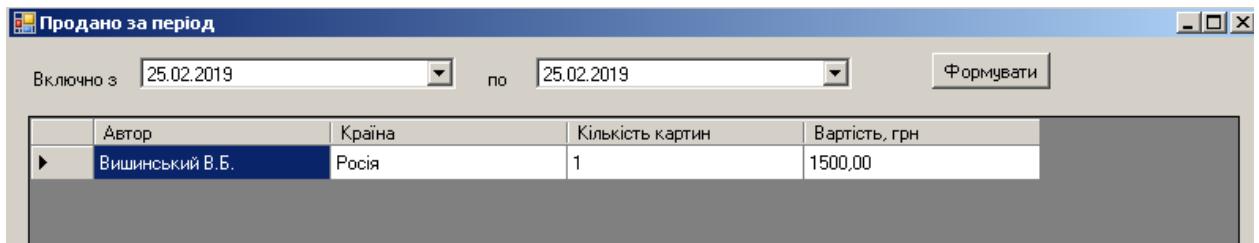
На рис. 5.30 показано звіт про експонати, що зараз знаходяться в експозиції. В цей звіт попадають лише приміщення, що у вікні довідника місць розташування на рис. 5.26 мають відповідну галочку "В показі".



Приміщення	Кількість картин	Кількість авторів	Кількість жанрів	Вартість, грн
зал № 1	1	1	1	9999,99
зал № 2	1	1	1	50000,00
зал № 3	2	1	1	3000,00

Рисунок 5.30 - Вікно звіту "В експозиції"

Звіт про продаж за період показує результати в розрізі авторів та країн, як видно на рис. 5.31. Обидві граничні дати включаються в період.



Автор	Країна	Кількість картин	Вартість, грн
Вишинський В.Б.	Росія	1	1500,00

Рисунок 5.31 - Вікно звіту "Продано за період"

На рис. 5.32 показано звіт про вартість експонатів по приміщенням. В цей звіт попадають лише приміщення, що у вікні довідника місць розташування на рис. 5.26 мають відповідну галочку "Наявність".

Вартість по приміщенням				
	Приміщення	Показ	Кількість картин	Вартість, грн
▶	зал № 1	в експозиції	1	9999,99
	зал № 2	в експозиції	1	50000,00
	зал № 3	в експозиції	2	3000,00
	склад № 2	зберігається	2	73800,00

Рисунок 5.32 - Вікно звіту "Вартість по приміщенням"

Звіт про кількість картин за країнами показує результати в розрізі країн та приміщень, як видно на рис. 5.33.

Кількість картин за країнами				
	Країна	Приміщення	Кількість картин	Вартість, грн
▶	Італія	зал № 1	1	9999,99
	Італія	зал № 3	2	3000,00
	Росія	склад № 2	1	3800,00
	Україна	зал № 2	1	50000,00
	Україна	склад № 2	1	70000,00

Рисунок 5.33 - Вікно звіту "Кількість картин за країнами"

Звіт про кількість картин за авторами показує результати в розрізі авторів та приміщень, як видно на рис. 5.34.

Кількість картин за авторами					
	Автор	Країна	Приміщення	Кількість картин	Вартість, грн
▶	Вишинський В.Б.	Росія	склад № 2	1	3800,00
	Леонардо да Вінчі	Італія	зал № 1	1	9999,99
	Рафаель Санті	Італія	зал № 3	2	3000,00
	Шевченко Т.Г.	Україна	зал № 2	1	50000,00
	Шевченко Т.Г.	Україна	склад № 2	1	70000,00

Рисунок 5.34 - Вікно звіту "Кількість картин за авторами"

Інформація про програму, показана на рис. 5.35, включає назву програми, дані автора, версію та рік.

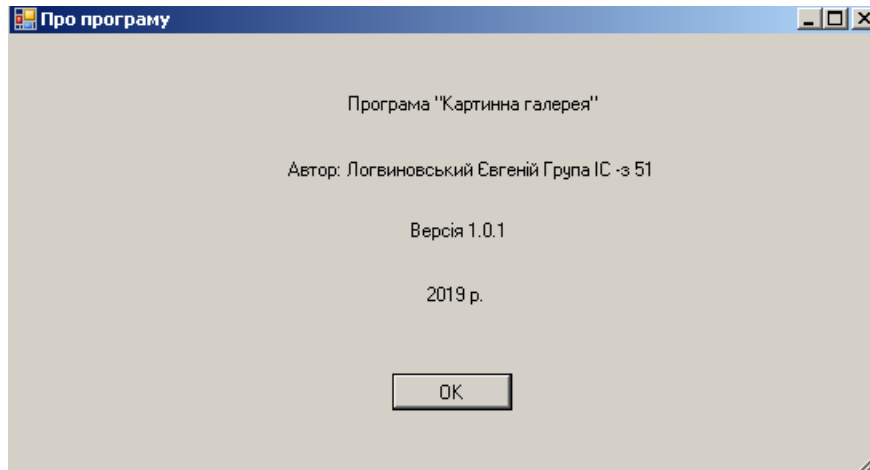


Рисунок 5.35 - Вікно інформації про програму

5.2 Випробування програмного продукту

5.2.1 Мета випробувань

Метою випробувань являється перевірка відповідності функцій комплексу задач з обліку експонатів картинної галереї вимогам технічного завдання.

5.2.2 Загальні положення

Випробування проводяться на основі наступних документів:

- ГОСТ 34.603–92. Інформаційна технологія. Види випробувань автоматизованих систем;
- ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

5.2.3 Результати випробувань

Результати випробувань наведені в табл. 5.1. Згідно отриманим результатам, програмне забезпечення комплексу задач повністю відповідає функціональним вимогам, представленим у технічному завданні на створення комплексу задач з обліку експонатів картинної галереї.

Таблиця 5.1

Опис тесту	Результат тесту	Оцінка
Введення логіну та паролю користувачів, що мають різні ролі	Деякі кнопки та меню стають недоступними, якщо роль користувача не адміністратор	Вірно
Введення помилкового логіну та паролю користувача	Програма завершує роботу	Вірно
Введення тексту замість числа, в поля рік та ціна	Програма виводить повідомлення про помилку	Вірно
За умови відсутності обраного рядка в таблиці натискання кнопок видалення та редагування	Програма виводить повідомлення про помилку, жодні дані не видаляються і не відкриваються у вікні редагування	Вірно
Введення експонату	Дані зберігаються в БД	Вірно
Видалення експонату	Видаляються і експонат з БД, і всі пов'язані з ним фотографії з файлового сервера	Вірно
Повторний вибір файлу фотографії для збереження	При копіюванні на файловий сервер до імені другого файлу фотографії дописується довільне число, що робить це ім'я унікальним в межах директорії	Вірно

Висновок до розділу

Програмне забезпечення комплексу задач повністю відповідає функціональним вимогам, представленим у технічному завданні на створення комплексу задач з обліку експонатів картинної галереї.

					ДП ІС-5104.1260-с.ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗАГАЛЬНІ ВИСНОВКИ

Мною розроблено комплекс задач з обліку експонатів картинної галереї. В процесі роботи мною на практиці пройдено основні етапи розробки програмного забезпечення.

Спочатку сформоване загальне уявлення про бізнес-процеси предметного середовища, що підлягають автоматизації. Проаналізована діяльність користувачів з різними обов'язками. Проведено огляд можливостей аналогічних програм, що представлені на ринку. Поставлені цілі та задачі, що мають бути вирішені для їх досягнення.

Далі визначено всю початкову інформацію, яка потрапляє до програми та всю звітну інформацію, яку програма видає в результаті роботи. На основі цієї інформації створено структуру БД у ЗНФ, яка повністю відповідає розробленому комплексу задач.

Створені в БД таблиці та процедури дозволяють отримати результати для відображення всіх необхідних звітів в табличній формі. Для формування запитів використовується мова SQL. Ці результати не потребують додаткової математичної обробки.

Потім за допомогою діаграм UML та словесного опису розглянуто вимоги до технічного забезпечення, обрану клієнт-серверну архітектуру програмного забезпечення, специфікації функцій клієнтської програми та склад звітів.

На останньому етапі створено програмне забезпечення та проведено його тестування.

Програмне забезпечення комплексу задач повністю відповідає функціональним вимогам, представленим у технічному завданні на створення комплексу задач з обліку експонатів картинної галереї.

ПЕРЕЛІК ПОСИЛАНЬ

1. Система управління складом [Електронний ресурс] — К.: 2019. — Режим доступу: <https://jsolutions.ua/ua/sistema-upravleniya-skladom>, вільний. — Заг. з екрану.
2. Мойсклад [Електронний ресурс] — М.: 2019. — Режим доступу: <https://www.moysklad.ru/programmy-dlja-biznesa/programma-businessman/>, вільний. — Заг. з екрану.
3. Програма складського обліку, програма для складу UniproRetail, складська програма для України [Електронний ресурс] — К.: 2019. — Режим доступу: <https://unipro.com.ua/ua/programa-skladskogo-obliku--programa-dlya-skladu-uniproretail--skladska-programa-dlya-ukrayiny/>, вільний. — Заг. з екрану.
4. Програма складського обліку [Електронний ресурс] — К.: 2019. — Режим доступу: https://remonline.ua/features/skladskoj_uchet/, вільний. — Заг. з екрану.
5. Укрсклад [Електронний ресурс] — К.: 2019. — Режим доступу: <https://www.ukrsklad.com/ua/home.html>, вільний. — Заг. з екрану.
6. Програма для складу [Електронний ресурс] — К.: 2019. — Режим доступу: http://ususoft.com.ua/uk/programma_dlya_sklada.php, вільний. — Заг. з екрану.
7. Storecalc [Електронний ресурс] — М.: 2019. — Режим доступу: <http://www.su-progs.com/langs/ukr/store.html>, вільний. — Заг. з екрану.
8. Складський облік [Електронний ресурс] — К.: 2019. — Режим доступу: <https://delovod.ua/uk/functionality/inventory>, вільний. — Заг. з екрану.
9. Мюллер Р. Д. Проектирование баз данных и UML. — М.: Лори, 2013. — 432 с.
10. Нестеров С.А. Базы данных. Учебник и практикум для академического бакалавриата. — М.: ЮРАЙТ, 2016. — 230 с.

11. Бураков П.В., Петров В.Ю. Введение в системы баз данных. Учебное пособие [Электронный ресурс].: — С-Пб., 2010. — 129 с.

12. Бен-Ган И. Microsoft SQL Server 2012. Основы T-SQL. — М.: Эксмо, 2017. — 401 с.

13. Гайдаржи В.І., Ізварін І.В. Бази даних в інформаційних системах. — К.: Университет "Україна", 2018. — 418 с.

14. Дейт К. SQL и реляционная теория. Как грамотно писать код на SQL. — М.: Символ-Плюс, 2010. — 480 с.

ДОДАТОК А

Тексти запитів для створення БД

Запити створення нової БД.

CREATE DATABASE [galereya] ON PRIMARY

(NAME = N'galereya', FILENAME = N'c:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\DATA\galereya.mdf' , SIZE = 4096KB , FILEGROWTH = 1024KB)

LOG ON

(NAME = N'galereya_log', FILENAME = N'c:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\DATA\galereya_log.ldf' , SIZE = 1024KB , FILEGROWTH = 10%)

GO

EXEC dbo.sp_dbcmptlevel @dbname=N'galereya', @new_cmptlevel=90

GO

ALTER DATABASE [galereya] SET ANSI_NULL_DEFAULT OFF

GO

ALTER DATABASE [galereya] SET ANSI_NULLS OFF

GO

ALTER DATABASE [galereya] SET ANSI_PADDING OFF

GO

ALTER DATABASE [galereya] SET ANSI_WARNINGS OFF

GO

ALTER DATABASE [galereya] SET ARITHABORT OFF

GO

ALTER DATABASE [galereya] SET AUTO_CLOSE OFF

GO

ALTER DATABASE [galereya] SET AUTO_CREATE_STATISTICS ON

GO

					ДП ІС-5104.1260-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		64

```
ALTER DATABASE [galereya] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [galereya] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [galereya] SET CURSOR_CLOSE_ON_COMMIT
OFF
GO
ALTER DATABASE [galereya] SET CURSOR_DEFAULT GLOBAL
GO
ALTER DATABASE [galereya] SET CONCAT_NULL_YIELDS_NULL
OFF
GO
ALTER DATABASE [galereya] SET NUMERIC_ROUNDABORT OFF
GO
ALTER DATABASE [galereya] SET QUOTED_IDENTIFIER OFF
GO
ALTER DATABASE [galereya] SET RECURSIVE_TRIGGERS OFF
GO
ALTER          DATABASE          [galereya]          SET
AUTO_UPDATE_STATISTICS_ASYNC OFF
GO
ALTER          DATABASE          [galereya]          SET
DATE_CORRELATION_OPTIMIZATION OFF
GO
ALTER DATABASE [galereya] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [galereya] SET READ_WRITE
GO
ALTER DATABASE [galereya] SET RECOVERY SIMPLE
```

GO

ALTER DATABASE [galereya] SET MULTI_USER

GO

ALTER DATABASE [galereya] SET PAGE_VERIFY CHECKSUM

GO

USE [galereya]

GO

IF NOT EXISTS (SELECT name FROM sys.filegroups WHERE
is_default=1 AND name = N'PRIMARY') ALTER DATABASE [galereya]
MODIFY FILEGROUP [PRIMARY] DEFAULT

GO

Запити створення користувача БД.

USE [master]

GO

CREATE LOGIN [galereyauser] WITH PASSWORD=N'galereyapass',
DEFAULT_DATABASE=[galereya], DEFAULT_LANGUAGE=[us_english],
CHECK_EXPIRATION=OFF, CHECK_POLICY=OFF

GO

ALTER LOGIN [galereyauser] ENABLE

GO

USE [galereya]

GO

CREATE USER [galereyausername] FOR LOGIN [galereyauser] WITH
DEFAULT_SCHEMA=[db_owner]

GO

USE [galereya]

GO

ALTER AUTHORIZATION ON SCHEMA::[db_accessadmin] TO
[galereyausername]

GO

USE [galereya]

GO

ALTER AUTHORIZATION ON SCHEMA::[db_securityadmin] TO
[galereyausername]

GO

USE [galereya]

GO

ALTER AUTHORIZATION ON SCHEMA::[db_owner] TO
[galereyausername]

GO

USE [galereya]

GO

ALTER AUTHORIZATION ON SCHEMA::[db_backupoperator] TO
[galereyausername]

GO

USE [galereya]

GO

ALTER AUTHORIZATION ON SCHEMA::[db_ddladmin] TO
[galereyausername]

GO

USE [galereya]

GO

ALTER AUTHORIZATION ON SCHEMA::[db_denydatawriter] TO
[galereyausername]

GO

USE [galereya]

```

GO
ALTER AUTHORIZATION ON SCHEMA::[db_datareader] TO
[galereyausername]
GO
USE [galereya]
GO
ALTER AUTHORIZATION ON SCHEMA::[db_datawriter] TO
[galereyausername]
GO
USE [galereya]
GO
ALTER AUTHORIZATION ON SCHEMA::[db_denydatareader] TO
[galereyausername]
GO
USE [galereya]
GO
EXEC sp_addrolemember N'db_securityadmin', N'galereyausername'
GO
USE [galereya]
GO
EXEC sp_addrolemember N'db_backupoperator', N'galereyausername'
GO
USE [galereya]
GO
EXEC sp_addrolemember N'db_datawriter', N'galereyausername'
GO
USE [galereya]
GO
EXEC sp_addrolemember N'db_accessadmin', N'galereyausername'

```

GO

USE [galereya]

GO

EXEC sp_addrolemember N'db_ddladmin', N'galereyausername'

GO

USE [galereya]

GO

EXEC sp_addrolemember N'db_owner', N'galereyausername'

GO

USE [galereya]

GO

EXEC sp_addrolemember N'db_datareader', N'galereyausername'

GO

Запити створення схеми для користувача БД.

ALTER DATABASE [galereya] COLLATE Cyrillic_General_CI_AS

GO

USE [galereya]

GO

CREATE SCHEMA galereya2 AUTHORIZATION galereyausername

GO

Запити для таблиці жанрів

USE [galereya]

GO

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

SET ANSI_PADDING ON

GO

```

CREATE TABLE [galereya2].[janr](
    [nrec] [decimal](20, 0) IDENTITY(1,1) NOT NULL,
    [janr_name] [varchar](50) NOT NULL,
    CONSTRAINT [PK_janr] PRIMARY KEY CLUSTERED
(
    [nrec] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
insert into galereya2.janr( janr_name) VALUES ('невідомий');
insert into galereya2.janr( janr_name) VALUES ('змішаний');
insert into galereya2.janr( janr_name) VALUES ('пейзаж');
insert into galereya2.janr( janr_name) VALUES ('портрет');
insert into galereya2.janr( janr_name) VALUES ('натюрморт');
insert into galereya2.janr( janr_name) VALUES ('історичний');
insert into galereya2.janr( janr_name) VALUES ('міфологічний');
insert into galereya2.janr( janr_name) VALUES ('батальний');
insert into galereya2.janr( janr_name) VALUES ('побутовий');
insert into galereya2.janr( janr_name) VALUES ('аніمالістичний');
GO
CREATE PROCEDURE [galereya2].[p_janr]
    -- процедура формування даних для випадającego списку
AS
BEGIN
    -- прибрати додаткові результати з запиту

```

SET NOCOUNT ON;

--запит

SELECT convert(decimal(20, 0), 0) as nrec, ' всі' as janr_name

UNION ALL

SELECT nrec, janr_name FROM janr

ORDER BY janr_name ;

END

GO

Запити для таблиці технік.

USE [galereya]

GO

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

SET ANSI_PADDING ON

GO

CREATE TABLE [galereya2].[tehnika](

[nrec] [decimal](20, 0) IDENTITY(1,1) NOT NULL,

[tehnika_name] [varchar](50) NOT NULL,

CONSTRAINT [PK_tehnika] PRIMARY KEY CLUSTERED

(

[nrec] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,

IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,

ALLOW_PAGE_LOCKS = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

SET ANSI_PADDING OFF

GO

```

insert into galereya2.tehnika( tehnika_name ) VALUES ('змішана' );
insert into galereya2.tehnika( tehnika_name ) VALUES ('фреска' );
insert into galereya2.tehnika( tehnika_name ) VALUES ('фотоdruk' );
insert into galereya2.tehnika( tehnika_name ) VALUES ('акварель' );
insert into galereya2.tehnika( tehnika_name ) VALUES ('туш' );
insert into galereya2.tehnika( tehnika_name ) VALUES ('карнація' );
insert into galereya2.tehnika( tehnika_name ) VALUES ('гризайль' );
insert into galereya2.tehnika( tehnika_name ) VALUES ('гуаш' );
insert into galereya2.tehnika( tehnika_name ) VALUES ('сфумато' );
insert into galereya2.tehnika( tehnika_name ) VALUES ('лесіровка' );
insert into galereya2.tehnika( tehnika_name ) VALUES ('пастель' );
insert into galereya2.tehnika( tehnika_name ) VALUES ('пуантилізм' );
insert into galereya2.tehnika( tehnika_name ) VALUES ('акріл' );

```

GO

CREATE PROCEDURE [galereya2].[p_tehnika]

-- процедура формування даних для випадającego списку

AS

BEGIN

-- прибрати додаткові результати з запиту

SET NOCOUNT ON;

--запит

SELECT convert(decimal(20, 0), 0) as nrec, ' всі' as tehnika_name

UNION ALL

SELECT nrec, tehnika_name FROM tehnika

ORDER BY tehnika_name ;

END

GO

Запити для таблиці ролей.

					ДП ІС-5104.1260-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		72

```

USE [galereya]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [galereya2].[rol](
    [nrec] [decimal](20, 0) NOT NULL,
    [rol_name] [varchar](50) NOT NULL,
    CONSTRAINT [PK_rol] PRIMARY KEY CLUSTERED
(
    [nrec] ASC
)WITH (PAD_INDEX = OFF, IGNORE_DUP_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
insert into galereya2.rol( nrec, rol_name) VALUES (1,'адміністратор');
insert into galereya2.rol( nrec, rol_name) VALUES (2,'менеджер');
insert into galereya2.rol( nrec, rol_name) VALUES (3,'комірник');
insert into galereya2.rol( nrec, rol_name) VALUES (4,'перегляд');
insert into galereya2.rol( nrec, rol_name) VALUES (5,'нема доступу');
GO
Запити для таблиці користувачів.
USE [galereya]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

```

```

CREATE FUNCTION galereya2.md5
(
    @value varchar(255)
)
RETURNS varchar(32)
AS
BEGIN
    RETURN SUBSTRING(sys.fn_sqlvarbasetostr(HASHBYTES('MD5',
@value)),3,32);
END
GO

CREATE TABLE [galereya2].[myuser](
    [nrec] [decimal](20, 0) IDENTITY(1,1) NOT NULL,
    [rol_nrec] [decimal](20, 0) NOT NULL,
    [d_begin] [datetime] NOT NULL,
    [d_end] [datetime] NOT NULL,
    [fio] [varchar](150) NOT NULL,
    [login] [varchar](50) NOT NULL,
    [pass] [varchar](32) NOT NULL,
    CONSTRAINT [PK_myuser] PRIMARY KEY CLUSTERED
(
    [nrec] ASC
)WITH (PAD_INDEX = OFF, IGNORE_DUP_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [galereya2].[myuser] WITH CHECK ADD
CONSTRAINT [FK_myuser_rol] FOREIGN KEY([rol_nrec])
REFERENCES [galereya2].[rol] ([nrec])

```

```

GO

ALTER TABLE [galereya2].[myuser] CHECK CONSTRAINT
[FK_myuser_rol]

GO

insert into galereya2.myuser( rol_nrec,d_begin,d_end,fio,login,pass)
VALUES (1,CAST('20190101' AS datetime), CAST('21991231' AS
datetime),'Адмінова А.Д.','1',galereya2.md5('1'));

insert into galereya2.myuser( rol_nrec,d_begin,d_end,fio,login,pass)
VALUES (2,CAST('20190101' AS datetime), CAST('21991231' AS
datetime),'Менеджеров М.Е.','2',galereya2.md5('2'));

insert into galereya2.myuser( rol_nrec,d_begin,d_end,fio,login,pass)
VALUES (3,CAST('20190101' AS datetime), CAST('21991231' AS
datetime),'Комірник К.О.','3',galereya2.md5('3'));

insert into galereya2.myuser( rol_nrec,d_begin,d_end,fio,login,pass)
VALUES (4,CAST('20190101' AS datetime), CAST('21991231' AS
datetime),'Переглядова П.Е.','4',galereya2.md5('4'));

GO

Запити для таблиці видів робіт

USE [galereya]

GO

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

SET ANSI_PADDING ON

GO

CREATE TABLE [galereya2].[robota](
    [nrec] [decimal](20, 0) IDENTITY(1,1) NOT NULL,
    [robota_name] [varchar](50) NOT NULL,

```

```

CONSTRAINT [PK_robota] PRIMARY KEY CLUSTERED
(
    [nrec] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
insert into galereya2.robota( robota_name) VALUES ('оригінал' );
insert into galereya2.robota( robota_name) VALUES ('репродукція' );
insert into galereya2.robota( robota_name) VALUES ('фотографія' );
GO
CREATE PROCEDURE [galereya2].[p_robota]
    -- процедура формування даних для випадającego списку
AS
BEGIN
    -- прибрати додаткові результати з запити
    SET NOCOUNT ON;
    --запит
    SELECT convert(decimal(20, 0), 0) as nrec, ' всі' as robota_name
    UNION ALL
    SELECT nrec, robota_name FROM robota
    ORDER BY robota_name ;
END
GO
Запити для таблиці основ.
USE [galereya]

```

```

GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [galereya2].[osnova](
    [nrec] [decimal](20, 0) IDENTITY(1,1) NOT NULL,
    [osnova_name] [varchar](50) NOT NULL,
    CONSTRAINT [PK_osnova] PRIMARY KEY CLUSTERED
(
    [nrec] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
insert into galereya2.osnova( osnova_name) VALUES ('змішана' );
insert into galereya2.osnova( osnova_name) VALUES ('холст' );
insert into galereya2.osnova( osnova_name) VALUES ('папір' );
insert into galereya2.osnova( osnova_name) VALUES ('дерево' );
insert into galereya2.osnova( osnova_name) VALUES ('фанера' );
insert into galereya2.osnova( osnova_name) VALUES ('пластик' );
insert into galereya2.osnova( osnova_name) VALUES ('камінь' );
insert into galereya2.osnova( osnova_name) VALUES ('метал' );
GO

```

```

CREATE PROCEDURE [galereya2].[p_osnova]
    -- процедура формування даних для випадającego списку
AS
BEGIN
    -- прибрати додаткові результати з запиту
    SET NOCOUNT ON;

    --запит
    SELECT convert(decimal(20, 0), 0) as nrec, ' всі' as osnova_name
    UNION ALL
    SELECT nrec, osnova_name FROM osnova
    ORDER BY osnova_name ;
END
GO

Запити для таблиці країн.
USE [galereya]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [galereya2].[kraina](
    [nrec] [decimal](20, 0) IDENTITY(1,1) NOT NULL,
    [kraina_name] [varchar](50) NOT NULL,
    CONSTRAINT [PK_kraina] PRIMARY KEY CLUSTERED
(
    [nrec] ASC

```

```

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = OFF) ON [PRIMARY]

```

```

) ON [PRIMARY]

```

```

GO

```

```

SET ANSI_PADDING OFF

```

```

GO

```

```

insert into galereya2.kraina( kraina_name) VALUES ('невідома' );

```

```

insert into galereya2.kraina( kraina_name) VALUES ('Треція' );

```

```

insert into galereya2.kraina( kraina_name) VALUES ('Італія' );

```

```

insert into galereya2.kraina( kraina_name) VALUES ('Франція' );

```

```

insert into galereya2.kraina( kraina_name) VALUES ('Україна' );

```

```

insert into galereya2.kraina( kraina_name) VALUES ('Росія' );

```

```

insert into galereya2.kraina( kraina_name) VALUES ('Білорусь' );

```

```

GO

```

```

CREATE PROCEDURE [galereya2].[p_kraina]

```

```

    -- процедура формування даних для випадяючого списку

```

```

AS

```

```

BEGIN

```

```

    -- прибрати додаткові результати з запиту

```

```

    SET NOCOUNT ON;

```

```

    --запит

```

```

    SELECT convert(decimal(20, 0), 0) as nrec, ' всі' as kraina_name

```

```

    UNION ALL

```

```

    SELECT nrec, kraina_name FROM kraina

```

```

    ORDER BY kraina_name ;

```

```

END

```

```

GO

```

Запити для таблиці авторів.


```

USE [galereya]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [galereya2].[avtor](
    [nrec] [decimal](20, 0) IDENTITY(1,1) NOT NULL,
    [avtor_pib] [varchar](150) NOT NULL,
    [kraina_nrec] [decimal](20, 0) NOT NULL,
    [avtor_prim] [varchar]( 500 ) NOT NULL,
    CONSTRAINT [PK_avtor] PRIMARY KEY CLUSTERED
(
    [nrec] ASC
)WITH (PAD_INDEX = OFF, IGNORE_DUP_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
ALTER TABLE [galereya2].[avtor] WITH CHECK ADD CONSTRAINT
[FK_avtor_kraina] FOREIGN KEY([kraina_nrec])
REFERENCES [galereya2].[kraina] ([nrec])
GO
ALTER TABLE [galereya2].[avtor] CHECK CONSTRAINT
[FK_vagon_kraina]
GO

insert into galereya2.avtor( avtor_pib, kraina_nrec, avtor_prim)
VALUES ('невідомий' , 1, 'невідомий');
insert into galereya2.avtor( avtor_pib, kraina_nrec, avtor_prim)

```

VALUES ('Мікеланджело Буонароті' , 3, 'Мікеланджело ді Франческо ді Нері ді Мініато дель Сера і Лодовіко ді Леонардо ді Буонарроті Сімоні (6 березня 1475, Капрезе — 18 лютого 1564, Рим)');

insert into galereya2.avtor(avtor_pib, kraina_nrec, avtor_prim)

VALUES ('Леонардо да Вінчі' , 3, '15 квітня 1452, селище Анк"яно, поблизу Флоренції — 2 травня 1519, замок Кло-Люсе, Амбуаз');

insert into galereya2.avtor(avtor_pib, kraina_nrec, avtor_prim)

VALUES ('Рафаель Санти' , 3, '28 березня чи 6 квітня 1483, Урбіно — 6 квітня 1520, Рим');

GO

CREATE PROCEDURE [galereya2].[p_avtor]

-- процедура формування даних для випадającego списку

AS

BEGIN

-- прибрати додаткові результати з запиту

SET NOCOUNT ON;

--запит

SELECT convert(decimal(20, 0), 0) as nrec, ' всі' as avtor_pib

UNION ALL

SELECT nrec, avtor_pib FROM avtor

ORDER BY avtor_pib ;

END

GO

Запити для таблиці картин.

USE [galereya]

GO

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [galereya2].[kartina](

[nrec] [decimal](20, 0) IDENTITY(1,1) NOT NULL,

[inv_nomer] [varchar](250) NOT NULL,

[nazva] [varchar](250) NOT NULL,

[rik] [INT] NOT NULL,

[cina] [decimal](20, 2) NOT NULL,

[rozmir_x] [decimal](5, 0) NOT NULL,

[rozmir_y] [decimal](5, 0) NOT NULL,

[janr_nrec] [decimal](20, 0) NOT NULL,

[tehnika_nrec] [decimal](20, 0) NULL,

[robota_nrec] [decimal](20, 0) NULL,

[osnova_nrec] [decimal](20, 0) NULL,

[avtor_nrec] [decimal](20, 0) NULL,

[kartina_prim] [varchar](500) NOT NULL,

CONSTRAINT [PK_kartina] PRIMARY KEY CLUSTERED

(

[nrec] ASC

)WITH (PAD_INDEX = OFF, IGNORE_DUP_KEY = OFF) ON

[PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [galereya2].[kartina] WITH CHECK ADD
CONSTRAINT [FK_kartina_janr] FOREIGN KEY([janr_nrec])

REFERENCES [galereya2].[janr] ([nrec]) ON DELETE CASCADE

GO

ALTER TABLE [galereya2].[kartina] CHECK CONSTRAINT
[FK_kartina_janr]

GO

```

ALTER TABLE [galereya2].[kartina] WITH CHECK ADD
CONSTRAINT [FK_kartina_tehnika] FOREIGN KEY([tehnika_nrec])
REFERENCES [galereya2].[tehnika] ([nrec])
GO
ALTER TABLE [galereya2].[kartina] CHECK CONSTRAINT
[FK_kartina_tehnika]
GO
ALTER TABLE [galereya2].[kartina] WITH CHECK ADD
CONSTRAINT [FK_kartina_robota] FOREIGN KEY([robota_nrec])
REFERENCES [galereya2].[robota] ([nrec])
GO
ALTER TABLE [galereya2].[kartina] CHECK CONSTRAINT
[FK_kartina_robota]
GO
ALTER TABLE [galereya2].[kartina] WITH CHECK ADD
CONSTRAINT [FK_kartina_osnova] FOREIGN KEY([osnova_nrec])
REFERENCES [galereya2].[osnova] ([nrec])
GO
ALTER TABLE [galereya2].[kartina] CHECK CONSTRAINT
[FK_kartina_osnova]
GO
ALTER TABLE [galereya2].[kartina] WITH CHECK ADD
CONSTRAINT [FK_kartina_avtor] FOREIGN KEY([avtor_nrec])
REFERENCES [galereya2].[avtor] ([nrec])
GO
ALTER TABLE [galereya2].[kartina] CHECK CONSTRAINT
[FK_kartina_avtor]
GO

```

```
insert into galereya2.kartina( inv_nomer, nazva, rik, cina, rozmir_x,
rozmir_y,
```

```
janr_nrec, tehnika_nrec, robota_nrec, osnova_nrec, avtor_nrec,
kartina_prim)
```

```
VALUES ('123-45/67', 'Мона Ліза', 1760, '9999.99', '100', '150', 4, 10, 2, 3,
3, 'примітка до картини');
```

```
insert into galereya2.kartina( inv_nomer, nazva, rik, cina, rozmir_x,
rozmir_y,
```

```
janr_nrec, tehnika_nrec, robota_nrec, osnova_nrec, avtor_nrec,
kartina_prim)
```

```
VALUES ('83-45/67', 'Мадона зі щігликом', 1785, '2000.00', '120', '200', 4,
12, 2, 3, 4, 'примітка до картини');
```

```
insert into galereya2.kartina( inv_nomer, nazva, rik, cina, rozmir_x,
rozmir_y,
```

```
janr_nrec, tehnika_nrec, robota_nrec, osnova_nrec, avtor_nrec,
kartina_prim)
```

```
VALUES ('44-45/67', 'Дама з єдинорогом', 1799, '1000.00', '50', '80', 4, 11,
2, 3, 4, 'примітка до картини');
```

Запити для таблиці місць розташування.

USE [galereya]

GO

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

SET ANSI_PADDING ON

GO

CREATE TABLE [galereya2].[misc](

[nrec] [decimal](20, 0) IDENTITY(1,1) NOT NULL,

```

[misce_name] [varchar](50) NOT NULL,
[nayavnist] [decimal](1, 0) NOT NULL,
[pokaz] [decimal](1, 0) NOT NULL,
CONSTRAINT [PK_misce] PRIMARY KEY CLUSTERED
(
    [nrec] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
insert into galereya2.misce( misce_name, nayavnist, pokaz) VALUES
('склад № 1', 1, 0);
insert into galereya2.misce( misce_name, nayavnist, pokaz) VALUES
('склад № 2', 1, 0);
insert into galereya2.misce( misce_name, nayavnist, pokaz) VALUES
('склад № 3', 1, 0);
insert into galereya2.misce( misce_name, nayavnist, pokaz) VALUES ('зал
№ 1', 1, 1);
insert into galereya2.misce( misce_name, nayavnist, pokaz) VALUES ('зал
№ 2', 1, 1);
insert into galereya2.misce( misce_name, nayavnist, pokaz) VALUES ('зал
№ 3', 1, 1);
insert into galereya2.misce( misce_name, nayavnist, pokaz) VALUES
('реставраційна майстерня', 1, 0);
insert into galereya2.misce( misce_name, nayavnist, pokaz) VALUES
('списана', 0, 0);

```

					ДП ІС-5104.1260-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		85

```
insert into galereya2.misce( misce_name, nayavnist, pokaz) VALUES
('викрадена', 0, 0);
```

```
insert into galereya2.misce( misce_name, nayavnist, pokaz) VALUES
('продана', 0, 0);
```

```
insert into galereya2.misce( misce_name, nayavnist, pokaz) VALUES
('тимчасово передана', 0, 0);
```

```
GO
```

```
CREATE PROCEDURE [galereya2].[p_misce]
```

```
-- процедура формування даних для випадającego списку
```

```
AS
```

```
BEGIN
```

```
-- прибрати додаткові результати з запиту
```

```
SET NOCOUNT ON;
```

```
--запит
```

```
SELECT convert(decimal(20, 0), 0) as nrec, ' всі' as misce_name
```

```
UNION ALL
```

```
SELECT nrec, misce_name FROM misce
```

```
ORDER BY misce_name ;
```

```
END
```

```
GO
```

```
Запити для таблиці руху.
```

```
USE [galereya]
```

```
GO
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE TABLE [galereya2].[ruh](
```

```
[nrec] [decimal](20, 0) IDENTITY(1,1) NOT NULL,
```

```

[merce_nrec] [decimal](20, 0) NOT NULL,
[kartina_nrec] [decimal](20, 0) NOT NULL,
[d_begin] [datetime] NOT NULL,
[d_end] [datetime] NULL,
[ruh_prim] [varchar](500) NOT NULL,
CONSTRAINT [PK_ruh] PRIMARY KEY CLUSTERED
(
    [nrec] ASC
)WITH (PAD_INDEX = OFF, IGNORE_DUP_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
ALTER TABLE [galereya2].[ruh] WITH CHECK ADD CONSTRAINT
[FK_ruh_misce] FOREIGN KEY([merce_nrec])
REFERENCES [galereya2].[merce] ([nrec])
GO
ALTER TABLE [galereya2].[ruh] CHECK CONSTRAINT [FK_ruh_misce]
GO
ALTER TABLE [galereya2].[ruh] WITH CHECK ADD CONSTRAINT
[FK_ruh_kartina] FOREIGN KEY([kartina_nrec])
REFERENCES [galereya2].[kartina] ([nrec])
GO
ALTER TABLE [galereya2].[ruh] CHECK CONSTRAINT
[FK_ruh_kartina]
GO
insert into galereya2.ruh( merce_nrec, kartina_nrec, d_begin, d_end,
ruh_prim)
VALUES (1, 2, CAST('20190101' AS datetime), CAST('20190118' AS
datetime) ,");

```



```

insert into galereya2.ruh( misce_nrec, kartina_nrec, d_begin, d_end,
ruh_prim)
VALUES (4, 2, CAST('20190118' AS datetime), null ,");
insert into galereya2.ruh( misce_nrec, kartina_nrec, d_begin, d_end,
ruh_prim)
VALUES (4, 1, CAST('20190215' AS datetime), null ,");
insert into galereya2.ruh( misce_nrec, kartina_nrec, d_begin, d_end,
ruh_prim)
VALUES (1, 3, CAST('20180830' AS datetime), CAST('20190109' AS
datetime) ,");
insert into galereya2.ruh( misce_nrec, kartina_nrec, d_begin, d_end,
ruh_prim)
VALUES (4, 3, CAST('20190109' AS datetime), CAST('20190220' AS
datetime) ,");
insert into galereya2.ruh( misce_nrec, kartina_nrec, d_begin, d_end,
ruh_prim)
VALUES (1, 3, CAST('20190220' AS datetime), null ,");
GO
CREATE FUNCTION galereya2.ostanniy_ruh
(
-- запуск наприклад select galereya2.ostanniy_ruh(1)
-- поверне ідентифікатор останнього руху картини з
ідентифікатором 1,
-- що треба для переносу цієї картини в інше місце
@kartina_nrec1 decimal(20, 0) -- картина
)
RETURNS decimal(20, 0)
AS
BEGIN

```

```

-- прибрати додаткові результати з запиту
--SET NOCOUNT ON;
declare @rez decimal(20, 0)
-- запит
SET @rez = (
    SELECT DISTINCT MAX(nrec) as tmp
    FROM galereya2.ruh
    WHERE kartina_nrec = @kartina_nrec1)
-- результат
RETURN @rez
END
GO
CREATE PROCEDURE [galereya2].[p_ruh]
-- процедура переміщення картини до нового місця
@misce_nrec1 decimal(20, 0), -- нове місце
@kartina_nrec1 decimal(20, 0), -- картина
@nrec1 decimal(20, 0) OUT -- результат = ідентифікатор місця
AS
BEGIN
    SET @nrec1 = 0;
-- прибрати додаткові результати з запиту
SET NOCOUNT ON;
--забрати з попереднього місця
UPDATE [galereya2].[ruh]
SET d_end = CONVERT (date, SYSDATETIME())
WHERE kartina_nrec = @kartina_nrec1 AND d_end IS NULL ;
-- перенести до нового місця
INSERT INTO [galereya2].[ruh] ( misce_nrec, kartina_nrec, d_begin,
d_end, ruh_prim)

```

					ДП ІС-5104.1260-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		89

```
VALUES (@misc_nrec1, @kartina_nrec1, CONVERT (date,
SYSDATETIME()), null ,");
```

```
--тест
```

```
SELECT @nrec1 = misc_nrec
```

```
FROM [galereya2].[ruh]
```

```
WHERE kartina_nrec = @kartina_nrec1 AND d_end IS NULL ;
```

```
-- результат
```

```
RETURN @nrec1
```

```
END
```

```
GO
```

Запити для таблиці фотографій.

```
USE [galereya]
```

```
GO
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE TABLE [galereya2].[foto](
```

```
    [nrec] [decimal](20, 0) IDENTITY(1,1) NOT NULL,
```

```
    [kartina_nrec] [decimal](20, 0) NOT NULL,
```

```
    [foto_dir] [varchar](250) NOT NULL,
```

```
    [foto_prim] [varchar](500) NOT NULL,
```

```
    CONSTRAINT [PK_foto] PRIMARY KEY CLUSTERED
```

```
(
```

```
    [nrec] ASC
```

```
)WITH (PAD_INDEX = OFF, IGNORE_DUP_KEY = OFF) ON
```

```
[PRIMARY]
```

```
) ON [PRIMARY]
```

```
GO
```

```
ALTER TABLE [galereya2].[foto] WITH CHECK ADD CONSTRAINT
[FK_foto_kartina] FOREIGN KEY([kartina_nrec])
```

```
REFERENCES [galereya2].[kartina] ([nrec])
```

```
GO
```

```
ALTER TABLE [galereya2].[foto] CHECK CONSTRAINT
[FK_foto_kartina]
```

```
GO
```

Запити створення процедури фільтрації списку експонатів.

```
USE [galereya]
```

```
GO
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE PROCEDURE [galereya2].[p_kartina]
```

```
-- Add the parameters for the stored procedure here
```

```
-- процедура формування даних про картини з параметрами:
```

```
@janr_nrec1 decimal(20, 0), --жанр
```

```
@tehnika_nrec1 decimal(20, 0), --техніка
```

```
@robota_nrec1 decimal(20, 0), --робота
```

```
@osnova_nrec1 decimal(20, 0), --основа
```

```
@kraina_nrec1 decimal(20, 0), --країна
```

```
@avtor_nrec1 decimal(20, 0), --автор
```

```
@misce_nrec1 decimal(20, 0), --місце
```

```
@nazva1 varchar(250), -- назва
```

```
@rik1 INT -- рік
```

```
AS
```

```
BEGIN
```

```
-- прибрати додаткові результати з SELECT
```

```
SET NOCOUNT ON;
```

```
--назва не пуста
```

```
DECLARE @nazva2 AS VARCHAR(250)
```

```
SET @nazva2 = isnull(@nazva1,")
```

```
IF @nazva2="
```

```
BEGIN
```

```
    SET @nazva2 = '%';
```

```
END
```

```
ELSE
```

```
BEGIN
```

```
    SET @nazva2 = '%'+@nazva2+'%';
```

```
END
```

```
--запит
```

```
SELECT
```

```
k.*,
```

```
j.janr_name,
```

```
t.tehnika_name,
```

```
r.robota_name,
```

```
o.osnova_name,
```

```
kr.kraina_name,
```

```
a.avtor_pib,
```

```
m.misce_name,
```

```
convert(varchar, ruh.d_begin, 104) as ruh_d_begin
```

```
FROM galereya2.kartina k, galereya2.janr j, galereya2.tehnika t,
```

```
galereya2.robota r, galereya2.osnova o,
```

```
galereya2.kraina kr, galereya2.avtor a , galereya2.misce m,
```

```
galereya2.ruh
```

```
WHERE
```

```
k.janr_nrec=j.nrec AND
```

					ДП ІС-5104.1260-с.ПЗ	Арк.
						92
Змн.	Арк.	№ докум.	Підпис	Дата		

```

k.tehnika_nrec=t.nrec AND
k.robota_nrec=r.nrec AND
k.osnova_nrec=o.nrec AND
a.kraina_nrec=kr.nrec AND
k.avtor_nrec=a.nrec AND
ruh.misce_nrec=m.nrec AND
ruh.kartina_nrec=k.nrec AND
ruh.d_end is null AND
( @janr_nrec1 = j.nrec OR @janr_nrec1 = 0) AND
( @tehnika_nrec1 = t.nrec OR @tehnika_nrec1 = 0) AND
( @robota_nrec1 = r.nrec OR @robota_nrec1 = 0) AND
( @osnova_nrec1 = o.nrec OR @osnova_nrec1 = 0) AND
( @kraina_nrec1 = kr.nrec OR @kraina_nrec1 = 0) AND
( @avtor_nrec1 = a.nrec OR @avtor_nrec1 = 0) AND
( @misce_nrec1 = m.nrec OR @misce_nrec1 = 0) AND
k.nazva LIKE @nazva2 AND
( @rik1 = k.rik OR @rik1 = 0)
ORDER BY k.nazva ;

END

GO

```

Додаток Б

Тексти програмного коду**Комплекс задач з обліку експонатів картинної галереї**

(Найменування програми (документа))

DVD-R

(Вид носія даних)

72 арк, 6453 кб

(Обсяг програми (документа) , арк.,) Кб)

Київ – 2019 року

					ДП ІС-5104.1260-с.ПЗ	Арк.
						94
Змн.	Арк.	№ докум.	Підпис	Дата		

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;
// для бази даних
using System.Configuration;
using System.Data.Odbc;
using System.Globalization;

namespace galereya
{
    public partial class Form1 : Form
    {
        private int user_rol; // роль користувача
        private int user_nrec; // ідентификатор користувача
        private string user_name; // ПІБ користувача
        private string connectionString; // рядок підключення до бази даних
        private string dir_foto; // директорія фото
        // параметри фільтру
        private int janr_nrec1; // жанр
        private int tehnika_nrec1; // техніка
        private int robota_nrec1; // робота
        private int osnova_nrec1; // основа
        private int kraina_nrec1; // країна
        private int avtor_nrec1; // автор

```



```
private int misce_nrec1 ;// місце знаходження
```

```
private string nazva1 ;// назва
```

```
private int rik1 ;// рік
```

```
public Form1()
```

```
{
```

```
    InitializeComponent();
```

```
}
```

```
public void obnovit_tab()
```

```
{
```

```
    // процедура формування даних про картини з параметрами:
```

```
    OdbcConnection cn;
```

```
    OdbcCommand cmd;
```

```
    DataSet dataSet = new DataSet();
```

```
    try
```

```
    {
```

```
        //підключитись до бази даних
```

```
        cn = new OdbcConnection(connectionString);
```

```
        cn.Open();
```

```
    }
```

```
    catch (OdbcException o)
```

```
    {
```

```
        MessageBox.Show(o.Message.ToString());
```

```
        return;
```

```
    }
```

```
    //отримати дані з бази даних
```

```
    cmd = new OdbcCommand("{call galereya2.p_kartina (?,?,?, ?,?,?, ?,?,?)}",
```

```
cn);
```

					ДП ІС-5104.1260-с.ПЗ	Арк.
						96
Змн.	Арк.	№ докум.	Підпис	Дата		

```

cmd.Parameters.AddWithValue("@janr_nrec1", janr_nrec1);
cmd.Parameters.AddWithValue("@tehnika_nrec1", tehnika_nrec1);
cmd.Parameters.AddWithValue("@robota_nrec1", robota_nrec1);
cmd.Parameters.AddWithValue("@osnova_nrec1", osnova_nrec1);
cmd.Parameters.AddWithValue("@kraina_nrec1", kraina_nrec1);
cmd.Parameters.AddWithValue("@avtor_nrec1", avtor_nrec1);
cmd.Parameters.AddWithValue("@misce_nrec1", misce_nrec1);
cmd.Parameters.AddWithValue("@nazva1", nazva1);
cmd.Parameters.AddWithValue("@rik1", rik1);
OdbcDataAdapter dataAdapter = new OdbcDataAdapter(cmd);
OdbcCommandBuilder commandBuilder = new
OdbcCommandBuilder(dataAdapter);
//показати дані в таблиці на екрані
DataTable table1 = new DataTable("tab1");
pkartinaBindingSource.DataSource = null;
dataSet.Tables.Add(table1);
table1.Locale = System.Globalization.CultureInfo.InvariantCulture;
dataAdapter.Fill(table1);
pkartinaBindingSource.DataSource = dataSet;
pkartinaBindingSource.DataMember = "tab1";
dataGridView1.ReadOnly = true;
dataGridView1.DataSource = pkartinaBindingSource;
}

private void vihidToolStripMenuItem_Click(object sender, EventArgs e)
{
//меню Вихід
Application.Exit();
}

```

```

private void Form1_Load(object sender, EventArgs e)
{
    //завантаження цієї форми
    string login, pass;
    // рядок підключення до бази даних
    connectionString =
ConfigurationManager.ConnectionStrings["galereya.Properties.Settings.galereyaC
onnectionString"].ConnectionString;
    //connectionString =
"Dsn=ms_sql_galereya;uid=galereyauser;pwd=galereyapass";
    user_rol = -1;
    user_nrec = -1;
    user_name = "невідомий";
    // показати форму авторизації
    Form2Login forma_vhid = new Form2Login();
    forma_vhid.ShowDialog();
    if (forma_vhid.DialogResult == DialogResult.OK)
    {
        // що введено користувачем
        login = forma_vhid.get_login();
        pass = forma_vhid.get_pass();
    }
    else
    {
        //вихід з програми
        Application.Exit();
        return;
    }
    //підключення до бази даних
    OdbcConnection conn = new OdbcConnection(connectionString);

```

					ДП ІС-5104.1260-с.ПЗ	Арк.
						98
Змн.	Арк.	№ докум.	Підпис	Дата		

```

conn.Open();
OdbcCommand myCommand = conn.CreateCommand();
// запит
myCommand.CommandText = "SELECT nrec, rol_nrec, fio FROM
[galereya].[galereya2].[myuser] " +
    " WHERE login=? and pass = galereya2.md5(?) and " +
    " GETDATE() between d_begin and d_end ";
// параметри запиту
myCommand.Parameters.AddWithValue("@login1", login);
myCommand.Parameters.AddWithValue("@pass1", pass);
// виконати запит
OdbcDataReader dataReader = myCommand.ExecuteReader();
if (dataReader.Read())
{
    user_nrec = Convert.ToInt32(dataReader["nrec"]);
    user_rol = Convert.ToInt32(dataReader["rol_nrec"]);
    user_name = dataReader["fio"].ToString();
}
dataReader.Close();
conn.Close();
//обмеження прав по ролям
switch (user_rol)
{
    case 1:
        //(1,'Адміністратор');
        //без обмежень
        break;
    case 2:
        //(2,'менеджер');
        //все, крім довідника користувачів

```

					ДП ІС-5104.1260-с.ПЗ	Арк.
						99
Змн.	Арк.	№ докум.	Підпис	Дата		

```

myuserToolStripMenuItem.Enabled = false;
break;
case 3:
    //(3,'копірник');
    //все, крім звітів та довідника користувачів
    myuserToolStripMenuItem.Enabled = false;
    zvitToolStripMenuItem.Enabled = false;
    break;
case 4:
    //(4,'перегляд');
    //лише допомога, перегляд фото та фільтр
    kartinaToolStripMenuItem.Enabled = false;
    dovidnikToolStripMenuItem.Enabled = false;
    zvitToolStripMenuItem.Enabled = false;
    button1.Enabled = false;
    button2.Enabled = false;
    button3.Enabled = false;
    button4.Enabled = false;
    break;
default:
    //нема прав
    MessageBox.Show("Ви не маєте доступу");
    //вихід з програми
    Application.Exit();
    break;
}

MessageBox.Show("Ви увійшли як " + user_name );
// параметри фільтру
janr_nrec1 =0;// жанр
tehnika_nrec1 =0;// техніка

```

					ДП ІС-5104.1260-с.ПЗ	Арк.
						100
Змн.	Арк.	№ докум.	Підпис	Дата		

```

robota_nrec1 =0;// робота
osnova_nrec1 =0;// основа
kraina_nrec1 =0;// країна
avtor_nrec1 =0;// автор
misce_nrec1 =0;// місце
nazva1 ="";// назва
rik1 =0;// рік

// для початку будуть показані всі картини
obnovit_tab();// процедура формування даних про картини
//прочитати файл параметрів
StreamReader fs = new StreamReader(@"param.ini");
dir_foto = fs.ReadLine();
}

```

```

private void button2_Click(object sender, EventArgs e)
{
    //кнопка Редагувати
    if ( dataGridView1.CurrentCell == null)
    {
        MessageBox.Show("Не обрано картину");
        return;
    }
    //передати обраний рядок в форму редагування
    Form6Edit f = new Form6Edit();
    string str;
    str =
dataGridView1.Rows[dataGridView1.CurrentCell.RowIndex].Cells[0].Value.ToSt
ring();
    f.nrec = Convert.ToInt32(str);
    f.r1 = dataGridView1.Rows[dataGridView1.CurrentCell.RowIndex];

```

					ДП ІС-5104.1260-с.ПЗ	Арк.
						101
Змн.	Арк.	№ докум.	Підпис	Дата		

```

f.ShowDialog();

//якщо користувач відмовився - нічого не міняти
if (f.DialogResult != DialogResult.OK)
{
    return;
}

//підключення до бази даних
OdbcConnection conn = new OdbcConnection(connectionString);
conn.Open();
OdbcCommand myCommand = conn.CreateCommand();

// запит
myCommand.CommandText =
    "UPDATE galereya2.kartina SET inv_nomer=?, nazva=?, rik=?, cina=?, " +
    " rozmir_x=?, rozmir_y=?, janr_nrec=?, tehnika_nrec=?, " +
    " robota_nrec=?, osnova_nrec=?, avtor_nrec=?, kartina_prim=? "+
    " WHERE nrec =?;";

// параметри запиту
myCommand.Parameters.AddWithValue("@inv_nomer1",
f.textBox1.Text.Trim());

myCommand.Parameters.AddWithValue("@nazva1", f.textBox3.Text.Trim());
myCommand.Parameters.AddWithValue("@rik1",
Convert.ToInt32(f.textBox4.Text.Trim()));

myCommand.Parameters.AddWithValue("@cina1",
Convert.ToDouble(f.textBox2.Text.Trim()));

myCommand.Parameters.AddWithValue("@rozmir_x1",
Convert.ToInt32(f.textBox5.Text.Trim()));

myCommand.Parameters.AddWithValue("@rozmir_y1",
Convert.ToInt32(f.textBox6.Text.Trim()));

myCommand.Parameters.AddWithValue("@janr_nrec1",
Convert.ToInt32(f.comboBox1.SelectedValue));

```

					ДП ІС-5104.1260-с.ПЗ	Арк.
						102
Змн.	Арк.	№ докум.	Підпис	Дата		

```

myCommand.Parameters.AddWithValue("@tehnika_nrec1",
Convert.ToInt32(f.comboBox2.SelectedValue));

myCommand.Parameters.AddWithValue("@robota_nrec1",
Convert.ToInt32(f.comboBox3.SelectedValue));

myCommand.Parameters.AddWithValue("@osnova_nrec1",
Convert.ToInt32(f.comboBox4.SelectedValue));

myCommand.Parameters.AddWithValue("@avtor_nrec1",
Convert.ToInt32(f.comboBox5.SelectedValue));

myCommand.Parameters.AddWithValue("@kartina_prim1",
f.textBox7.Text.Trim());

myCommand.Parameters.AddWithValue("@nrec1", f.nrec );

// виконати запит
myCommand.ExecuteNonQuery();

conn.Close();

obnovit_tab(); // процедура формування даних про картини
}

```

```

private void button1_Click(object sender, EventArgs e)
{
    //кнопка Створити
    //форма редагування
    Form6Edit f = new Form6Edit();
    f.nrec = 0; //новий рядок
    f.ShowDialog();

    //якщо користувач відмовився - нічого не міняти
    if (f.DialogResult != DialogResult.OK)
    {
        return;
    }

    //підключення до бази даних

```



```

OdbcConnection conn = new OdbcConnection(connectionString);
conn.Open();
OdbcCommand myCommand = conn.CreateCommand();
//початок транзакції
myCommand.CommandText = "begin transaction;";
myCommand.CommandType = CommandType.Text;
myCommand.ExecuteNonQuery();
// запит
myCommand.CommandText =
    "insert into galereya2.kartina( inv_nomer, nazva, rik, cina, rozmir_x,
rozmir_y," +
    " janr_nrec, tehnika_nrec, robota_nrec, osnova_nrec, avtor_nrec,
kartina_prim) " +
    " VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?);" +
    " SELECT SCOPE_IDENTITY();";
// параметри запиту
myCommand.Parameters.AddWithValue("@inv_nomer1",
f.textBox1.Text.Trim());
myCommand.Parameters.AddWithValue("@nazva1", f.textBox3.Text.Trim());
myCommand.Parameters.AddWithValue("@rik1",
Convert.ToInt32(f.textBox4.Text.Trim()));
myCommand.Parameters.AddWithValue("@cina1", Convert.ToDouble(
f.textBox2.Text.Trim()));
myCommand.Parameters.AddWithValue("@rozmir_x1",
Convert.ToInt32(f.textBox5.Text.Trim()));
myCommand.Parameters.AddWithValue("@rozmir_y1",
Convert.ToInt32(f.textBox6.Text.Trim()));
myCommand.Parameters.AddWithValue("@janr_nrec1",
Convert.ToInt32(f.comboBox1.SelectedValue));

```

```

myCommand.Parameters.AddWithValue("@tehnika_nrec1",
Convert.ToInt32(f.comboBox2.SelectedValue));

myCommand.Parameters.AddWithValue("@robota_nrec1",
Convert.ToInt32(f.comboBox3.SelectedValue));

myCommand.Parameters.AddWithValue("@osnova_nrec1",
Convert.ToInt32(f.comboBox4.SelectedValue));

myCommand.Parameters.AddWithValue("@avtor_nrec1",
Convert.ToInt32(f.comboBox5.SelectedValue));

myCommand.Parameters.AddWithValue("@kartina_prim1",
f.textBox7.Text.Trim());

// виконати запит
int kartina_nrec, ruh_nrec=0;
kartina_nrec = Convert.ToInt32(myCommand.ExecuteScalar());

//розмістити картину
myCommand = new OdbcCommand("{ call galereya2.p_ruh (?, ?, ?) }", conn);
myCommand.CommandType = CommandType.StoredProcedure;

// параметри запиту
myCommand.Parameters.AddWithValue("@misce_nrec1",
Convert.ToInt32(f.comboBox6.SelectedValue));

myCommand.Parameters.AddWithValue("@kartina_nrec1", kartina_nrec);
myCommand.Parameters.AddWithValue("@nrec1", ruh_nrec).Direction =
ParameterDirection.Output;

myCommand.ExecuteNonQuery();

int ruh_nrec2 = Convert.ToInt32(myCommand.Parameters["@nrec1"].Value);

// завершити транзакцію
myCommand.CommandText = "commit;";
myCommand.CommandType = CommandType.Text;
myCommand.ExecuteNonQuery();

conn.Close();

//MessageBox.Show(ruh_nrec2.ToString());

```

```

    obnovit_tab(); // процедура формування даних про картини
}

private void button3_Click(object sender, EventArgs e)
{
    //кнопка Видалити
    if (dataGridView1.SelectedRows.Count < 1)
    {
        MessageBox.Show("Не обрані рядки для видалення.");
    }
    else
    {
        string str;
        int kartina_nrec;
        //підключення до бази даних
        OdbcConnection conn = new OdbcConnection(connectionString);
        conn.Open();
        //видалити всі виділені рядки
        foreach (DataGridViewRow row in dataGridView1.SelectedRows)
        {
            if (!row.IsNewRow)
            {
                str = row.Cells[0].Value.ToString(); //nrec
                kartina_nrec = Convert.ToInt32(str);
                //спочатку видалення файлів
                delete_files(kartina_nrec);
                //видалення з дочірніх таблиць
                // запит
                OdbcCommand myCommand = conn.CreateCommand();
                myCommand.CommandType = CommandType.Text;

```

					ДП ІС-5104.1260-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		106

```

myCommand.CommandText =
    "DELETE FROM galereya2.ruh " +
    " WHERE kartina_nrec = ?;";
// параметри запиту
myCommand.Parameters.AddWithValue("@nrec1", kartina_nrec);
myCommand.ExecuteNonQuery();
// запит
myCommand.CommandText =
    "DELETE FROM galereya2.foto " +
    " WHERE kartina_nrec = ?;";
// параметри запиту
myCommand.Parameters.AddWithValue("@nrec1", kartina_nrec);
myCommand.ExecuteNonQuery();
//виделення з батьківської таблиці
// запит
myCommand.CommandText =
    "DELETE FROM galereya2.kartina " +
    " WHERE nrec = ?;";
// параметри запиту
myCommand.Parameters.AddWithValue("@nrec1", kartina_nrec);
myCommand.ExecuteNonQuery();
MessageBox.Show("Видалено картину з ідентифікатором " + str);
}
}
conn.Close();
obnovit_tab(); // процедура формування даних про картини
}
}

```

```
private void button4_Click(object sender, EventArgs e)
```

					ДП ІС-5104.1260-с.ПЗ	Арк.
						107
Змн.	Арк.	№ докум.	Підпис	Дата		

```

{
    //кнопка Перемістити
    if (dataGridView1.CurrentCell == null)
    {
        MessageBox.Show("Не обрано картину");
        return;
    }
    //обраний рядок
    string str;
    str =
dataGridView1.Rows[dataGridView1.CurrentCell.RowIndex].Cells[0].Value.ToSt
ring();
    int kartina_nrec = Convert.ToInt32(str);
    //форма вибору місця
    Form7Ruh f = new Form7Ruh();
    f.ShowDialog();
    //якщо користувач відмовився - нічого не міняти
    if (f.DialogResult != DialogResult.OK)
    {
        return;
    }
    int misce_nrec = f.get_misce(); // нове місце знаходження
    //підключення до бази даних
    OdbcConnection conn = new OdbcConnection(connectionString);
    conn.Open();
    OdbcCommand myCommand = conn.CreateCommand();
    // виконати запит
    int ruh_nrec = 0;
    //розмістити картину
    myCommand = new OdbcCommand("{call galereya2.p_ruh (?,?,?)}", conn);

```

					ДП ІС-5104.1260-с.ПЗ	Арк.
						108
Змн.	Арк.	№ докум.	Підпис	Дата		

```

myCommand.CommandType = CommandType.StoredProcedure;
// параметри запиту
myCommand.Parameters.AddWithValue("@misc_nrec1", misc_nrec);
myCommand.Parameters.AddWithValue("@kartina_nrec1", kartina_nrec);
myCommand.Parameters.AddWithValue("@nrec1", ruh_nrec).Direction =
ParameterDirection.Output;
myCommand.ExecuteNonQuery();
int ruh_nrec2 = Convert.ToInt32(myCommand.Parameters["@nrec1"].Value);
conn.Close();
//MessageBox.Show(ruh_nrec2.ToString());
obnovit_tab(); // процедура формування даних про картини
}

private void button5_Click(object sender, EventArgs e)
{
    //кнопка Фото
    if (dataGridView1.CurrentCell == null)
    {
        MessageBox.Show("Не обрано картину");
        return;
    }
    //обраний рядок
    string str;
    str =
dataGridView1.Rows[dataGridView1.CurrentCell.RowIndex].Cells[0].Value.ToSt
ring();
    int kartina_nrec = Convert.ToInt32(str);
    //форма редагування
    Form8Foto f = new Form8Foto();
    f.set_parametri( dir_foto, kartina_nrec, user_rol);

```

```

f.ShowDialog();
}

private void button6_Click(object sender, EventArgs e)
{
    //кнопка Фільтр
    Form4Filtr f = new Form4Filtr();
    f.ShowDialog();
    //якщо користувач відмовився - нічого не міняти
    if (f.DialogResult != DialogResult.OK)
    {
        return;
    }
    // отримання вибору користувача
    janr_nrec1 = f.get_janr();// жанр
    tehnika_nrec1 = f.get_tehnika();// техніка
    robota_nrec1 = f.get_robota();// робота
    osnova_nrec1 = f.get_osnova();// основа
    kraina_nrec1 = f.get_kraina();// країна
    avtor_nrec1 = f.get_avtor();// автор
    misce_nrec1 = f.get_misce();// місце
    nazva1 = f.get_nazva();// назва
    rik1 = f.get_rik();//рік
    obnovit_tab();// процедура формування даних про картини
}

private void myuserToolStripMenuItem_Click(object sender, EventArgs e)
{
    // меню Користувач
    Form3User f = new Form3User();

```

					ДП ІС-5104.1260-с.ПЗ	Арк.
						110
Змн.	Арк.	№ докум.	Підпис	Дата		

```
f.connectionString = connectionString;
f.ShowDialog();
}
```

```
private void aboutToolStripMenuItem_Click(object sender, EventArgs e)
{
    //меню Про програму
    Form5About f = new Form5About();
    f.ShowDialog();
}
```

```
private void k_newToolStripMenuItem_Click(object sender, EventArgs e)
{
    //меню Створити
    button1.PerformClick();
}
```

```
private void k_editToolStripMenuItem_Click(object sender, EventArgs e)
{
    //меню Редагувати
    button2.PerformClick();
}
```

```
private void k_delToolStripMenuItem_Click(object sender, EventArgs e)
{
    //меню Видалити
    button3.PerformClick();
}
```

```
private void k_ruhToolStripMenuItem_Click(object sender, EventArgs e)
```

					ДП ІС-5104.1260-с.ПЗ	Арк.
						111
Змн.	Арк.	№ докум.	Підпис	Дата		


```
{
    //меню Перемістити
    button4.PerformClick();
}
```

```
private void k_fotoToolStripMenuItem_Click(object sender, EventArgs e)
{
    //меню Фото
    button5.PerformClick();
}
```

```
private void k_filtrToolStripMenuItem_Click(object sender, EventArgs e)
{
    //меню Фільтр
    button6.PerformClick();
}
```

```
private void delete_files(int kartina_nrec)
{
    //видалення всіх файлів фото картини
    string fn;
    //підключення до бази даних
    OdbcConnection conn = new OdbcConnection(connectionString);
    conn.Open();
    OdbcCommand myCommand = conn.CreateCommand();
    // запит
    myCommand.CommandText = "SELECT foto_dir FROM
[galereya].[galereya2].[foto] " +
    " WHERE kartina_nrec=? ;";
```

```

// параметри запиту
myCommand.Parameters.AddWithValue("@kartina_nrec1", kartina_nrec);
// виконати запит
OdbcDataReader dataReader = myCommand.ExecuteReader();
if (dataReader.HasRows)
{
    while (dataReader.Read())
    {
        //видалити файл
        fn = dir_foto + dataReader["foto_dir"].ToString();
        try
        {
            File.Delete(fn);
            MessageBox.Show("Виделено файл " + fn);
        }
        catch
        {
            MessageBox.Show("Помилка виделення файла " + fn);
        }
    }
}
dataReader.Close();
conn.Close();
}

private void misceToolStripMenuItem_Click(object sender, EventArgs e)
{
    //меню Місце
    Form15Misce f = new Form15Misce();
    f.ShowDialog();
}

```

```

    obnovit_tab(); // процедура формування даних про картини
}

private void avtorToolStripMenuItem_Click(object sender, EventArgs e)
{
    //меню Автор
    Form14avtor f = new Form14avtor();
    f.ShowDialog();
    obnovit_tab(); // процедура формування даних про картини
}

private void krainaToolStripMenuItem_Click(object sender, EventArgs e)
{
    //меню Країна
    Form13Kraina f = new Form13Kraina();
    f.ShowDialog();
    obnovit_tab(); // процедура формування даних про картини
}

private void osnovaToolStripMenuItem_Click(object sender, EventArgs e)
{
    //меню Основа
    Form12Osnova f = new Form12Osnova();
    f.ShowDialog();
    obnovit_tab(); // процедура формування даних про картини
}

private void robotaToolStripMenuItem_Click(object sender, EventArgs e)
{
    //меню Робота

```

					ДП ІС-5104.1260-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		114

```

Form11Robota f = new Form11Robota();
f.ShowDialog();
obnovit_tab(); // процедура формування даних про картини
}

private void tehnikaToolStripMenuItem_Click(object sender, EventArgs e)
{
    //меню Техника
    Form10Tehnika f = new Form10Tehnika();
    f.ShowDialog();
    obnovit_tab(); // процедура формування даних про картини
}

private void janrToolStripMenuItem_Click(object sender, EventArgs e)
{
    //меню Жанр
    Form9Janr f = new Form9Janr();
    f.ShowDialog();
    obnovit_tab(); // процедура формування даних про картини
}

private void zpokazToolStripMenuItem_Click(object sender, EventArgs e)
{
    //меню В експозиції
    Form16Z_ekspoziciya f = new Form16Z_ekspoziciya();
    f.pokaz();
    f.ShowDialog();
}

private void zprodanoToolStripMenuItem_Click(object sender, EventArgs e)

```

					ДП ІС-5104.1260-с.ПЗ	Арк.
						115
Змн.	Арк.	№ докум.	Підпис	Дата		

```

{
    //меню Продано за період
    Form17Z_prodano f = new Form17Z_prodano();
    f.sogodni();
    f.pokaz();
    f.ShowDialog();
}

private void zvariantToolStripMenuItem_Click(object sender, EventArgs e)
{
    //меню Вартість по приміщенням
    Form18Z_variant f = new Form18Z_variant();
    f.pokaz();
    f.ShowDialog();
}

private void zkrainaToolStripMenuItem_Click(object sender, EventArgs e)
{
    //меню Кількість картин за країнами
    Form19Z_kilkist_kraina f = new Form19Z_kilkist_kraina();
    f.pokaz();
    f.ShowDialog();
}

private void zavtorToolStripMenuItem_Click(object sender, EventArgs e)
{
    //меню Кількість картин за авторами
    Form20Z_kilkist_avtor f = new Form20Z_kilkist_avtor();
    f.pokaz();
    f.ShowDialog();
}

```

```

    }
}
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace galereya
{
    public partial class Form10Tehnika : Form
    {
        public Form10Tehnika()
        {
            InitializeComponent();

            private void button3_Click(object sender, EventArgs e)
            {
                //Зберегти зміни
                this.tehnikaTableAdapter.Update(this.galereyaDataSet.tehnika);
            }

            private void button2_Click(object sender, EventArgs e)
            {
                //видалити рядок

```

					ДП ІС-5104.1260-с.ПЗ	Арк.
						117
Змн.	Арк.	№ докум.	Підпис	Дата		

```

if (dataGridView1.SelectedRows.Count < 1)
{
    MessageBox.Show("Не обрані рядки для видалення.");
}
else
{
    //видалити всі виділені рядки
    foreach (DataGridViewRow row in dataGridView1.SelectedRows)
    {
        if (!row.IsNewRow) dataGridView1.Rows.Remove(row);
    }
}

}

private void button1_Click(object sender, EventArgs e)
{
    //Створити рядок
    this.tehnikaTableAdapter.Insert("введіть");
    //внести в базу зміни
    tehnikaTableAdapter.Update(galereyaDataSet);
    //оновити дані в формі
    this.tehnikaTableAdapter.Fill(this.galereyaDataSet.tehnika);
    int row1 = dataGridView1.RowCount - 1;
    if (row1 > -1)
    {
        dataGridView1.FirstDisplayedScrollingRowIndex = row1;
        DataGridViewCell cell = dataGridView1.Rows[row1].Cells[1];
        dataGridView1.CurrentCell = cell;
        dataGridView1.CurrentCell.Selected = true;
    }
}

```

					ДП ІС-5104.1260-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		118

}

}

```
private void Form10Tehnika_Load(object sender, EventArgs e)
```

{

```
// TODO: This line of code loads data into the 'galereyaDataSet.tehnika' table.
```

You can move, or remove it, as needed.

```
this.tehnikaTableAdapter.Fill(this.galereyaDataSet.tehnika);
```

}

}

}

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.ComponentModel;
```

```
using System.Data;
```

```
using System.Drawing;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Windows.Forms;
```

```
namespace galereya
```

{

```
public partial class Form11Robota : Form
```

{

```
public Form11Robota()
```

{

```
InitializeComponent();
```

}

```
private void button3_Click(object sender, EventArgs e)
```



```

{
    //Зберегти зміни
    this.robotaTableAdapter.Update(this.galereyaDataSet.robota);
}

private void button2_Click(object sender, EventArgs e)
{
    //видалити рядок
    if (dataGridView1.SelectedRows.Count < 1)
    {
        MessageBox.Show("Не обрані рядки для видалення.");
    }
    else
    {
        //видалити всі виділені рядки
        foreach (DataGridViewRow row in dataGridView1.SelectedRows)
        {
            if (!row.IsNewRow) dataGridView1.Rows.Remove(row);
        }
    }
}

private void button1_Click(object sender, EventArgs e)
{
    //Створити рядок
    this.robotaTableAdapter.Insert("введіть");
    //внести в базу зміни
    robotaTableAdapter.Update(galereyaDataSet);
    //оновити дані в формі
    this.robotaTableAdapter.Fill(this.galereyaDataSet.robota);
}

```

					ДП ІС-5104.1260-с.ПЗ	Арк.
						120
Змн.	Арк.	№ докум.	Підпис	Дата		

```

int row1 = dataGridView1.RowCount - 1;
if (row1 > -1)
{
    dataGridView1.FirstDisplayedScrollingRowIndex = row1;
    DataGridViewCell cell = dataGridView1.Rows[row1].Cells[1];
    dataGridView1.CurrentCell = cell;
    dataGridView1.CurrentCell.Selected = true;
}
}

```

```

private void Form11Robota_Load(object sender, EventArgs e)

```

```

{

```

```

    // TODO: This line of code loads data into the 'galereyaDataSet.robota' table.

```

You can move, or remove it, as needed.

```

    this.robotaTableAdapter.Fill(this.galereyaDataSet.robota);

```

```

}

```

```

}

```

```

}

```

```

using System;

```

```

using System.Collections.Generic;

```

```

using System.ComponentModel;

```

```

using System.Data;

```

```

using System.Drawing;

```

```

using System.Linq;

```

```

using System.Text;

```

```

using System.Windows.Forms;

```

```

namespace galereya

```

```

{

```

```

    public partial class Form12Osnova : Form

```

					ДП ІС-5104.1260-с.ПЗ	Арк.
						121
Змн.	Арк.	№ докум.	Підпис	Дата		

```

{
    public Form12Osnova()
    {
        InitializeComponent();
    }

    private void button3_Click(object sender, EventArgs e)
    {
        //Зберегти зміни
        this.osnovaTableAdapter.Update(this.galereyaDataSet.osnova);
    }

    private void button2_Click(object sender, EventArgs e)
    {
        //видалити рядок
        if (dataGridView1.SelectedRows.Count < 1)
        {
            MessageBox.Show("Не обрані рядки для видалення.");
        }
        else
        {
            //видалити всі виділені рядки
            foreach (DataGridViewRow row in dataGridView1.SelectedRows)
            {
                if (!row.IsNewRow) dataGridView1.Rows.Remove(row);
            }
        }
    }
}

```

```

private void button1_Click(object sender, EventArgs e)
{
    //Створити рядок
    this.osnovaTableAdapter.Insert("введіть");
    //внести в базу зміни
    osnovaTableAdapter.Update(galereyaDataSet);
    //оновити дані в формі
    this.osnovaTableAdapter.Fill(this.galereyaDataSet.osnova);
    int row1 = dataGridView1.RowCount - 1;
    if (row1 > -1)
    {
        dataGridView1.FirstDisplayedScrollingRowIndex = row1;
        DataGridViewCell cell = dataGridView1.Rows[row1].Cells[1];
        dataGridView1.CurrentCell = cell;
        dataGridView1.CurrentCell.Selected = true;
    }
}

```

```

private void Form12Osnova_Load(object sender, EventArgs e)
{

```

// TODO: This line of code loads data into the 'galereyaDataSet.osnova' table.

You can move, or remove it, as needed.

```

    this.osnovaTableAdapter.Fill(this.galereyaDataSet.osnova);

```

```

    }

```

```

}

```

```

}

```

```

using System;

```

```

using System.Collections.Generic;

```

```

using System.ComponentModel;

```

					ДП ІС-5104.1260-с.ПЗ	Арк.
						123
Змн.	Арк.	№ докум.	Підпис	Дата		

```

using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace galereya
{
    public partial class Form13Kraina : Form
    {
        public Form13Kraina()
        {
            InitializeComponent();

            private void button3_Click(object sender, EventArgs e)
            {
                //Зберегти зміни
                this.krainaTableAdapter.Update(this.galereyaDataSet.kraina);
            }

            private void button1_Click(object sender, EventArgs e)
            {
                //Створити рядок
                this.krainaTableAdapter.Insert( "введіть");
                //внести в базу зміни
                krainaTableAdapter.Update(galereyaDataSet);
                //оновити дані в формі
                this.krainaTableAdapter.Fill(this.galereyaDataSet.kraina);
                int row1 = dataGridView1.RowCount - 1;

```

```

if (row1 > -1)
{
    dataGridView1.FirstDisplayedScrollingRowIndex = row1;
    DataGridViewCell cell = dataGridView1.Rows[row1].Cells[1];
    dataGridView1.CurrentCell = cell;
    dataGridView1.CurrentCell.Selected = true;
}
}

```

```

private void button2_Click(object sender, EventArgs e)
{
    //видалити рядок
    if (dataGridView1.SelectedRows.Count < 1)
    {
        MessageBox.Show("Не обрані рядки для видалення.");
    }
    else
    {
        //видалити всі виділені рядки
        foreach (DataGridViewRow row in dataGridView1.SelectedRows)
        {
            if (!row.IsNewRow) dataGridView1.Rows.Remove(row);
        }
    }
}

```

```

private void Form13Kraina_Load(object sender, EventArgs e)
{
    // TODO: This line of code loads data into the 'galereyaDataSet.kraina' table.

```

You can move, or remove it, as needed.

					ДП ІС-5104.1260-с.ПЗ	Арк.
						125
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        this.krainaTableAdapter.Fill(this.galereyaDataSet.kraina);
    }
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace galereya
{
    public partial class Form14avtor : Form
    {
        public Form14avtor()
        {
            InitializeComponent();
        }

        private void Form14avtor_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the 'galereyaDataSet.kraina' table.
            You can move, or remove it, as needed.

            this.krainaTableAdapter.Fill(this.galereyaDataSet.kraina);

            // TODO: This line of code loads data into the 'galereyaDataSet.avtor' table.
            You can move, or remove it, as needed.

            this.avtorTableAdapter.Fill(this.galereyaDataSet.avtor);

```

					ДП ІС-5104.1260-с.ПЗ	Арк.
						126
Змн.	Арк.	№ докум.	Підпис	Дата		

}

```
private void button1_Click(object sender, EventArgs e)
```

{

```
//Створити рядок
```

```
// обрати дійсне значення ідентифікатора країни
```

```
int kraina_nrec;
```

```
kraina_nrec =
```

```
Convert.ToInt32(galereyaDataSet.kraina.Rows[0].ItemArray[0].ToString());
```

```
this.avtorTableAdapter.Insert("введіть", kraina_nrec, "");
```

```
//внести в базу зміни
```

```
avtorTableAdapter.Update(galereyaDataSet);
```

```
//оновити дані в формі
```

```
this.avtorTableAdapter.Fill(this.galereyaDataSet.avtor);
```

```
int row1 = dataGridView1.RowCount - 1;
```

```
if (row1 > -1)
```

{

```
dataGridView1.FirstDisplayedScrollingRowIndex = row1;
```

```
DataGridViewCell cell = dataGridView1.Rows[row1].Cells[1];
```

```
dataGridView1.CurrentCell = cell;
```

```
dataGridView1.CurrentCell.Selected = true;
```

}

}

```
private void button2_Click(object sender, EventArgs e)
```

{

```
//видалити рядок
```

```
if (dataGridView1.SelectedRows.Count < 1)
```

{

```
MessageBox.Show("Не обрані рядки для видалення.");
```



```

    }
    else
    {
        //видалити всі виділені рядки
        foreach (DataGridViewRow row in dataGridView1.SelectedRows)
        {
            if (!row.IsNewRow) dataGridView1.Rows.Remove(row);
        }
    }
}

```

```

private void button3_Click(object sender, EventArgs e)
{
    //Зберегти зміни
    this.avtorTableAdapter.Update(this.galereyaDataSet.avtor);
}
}
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

```

```
namespace galereya
```

```
{
```

```
public partial class Form15Misce : Form
```

					ДП ІС-5104.1260-с.ПЗ	Арк.
						128
Змн.	Арк.	№ докум.	Підпис	Дата		

```

{
    public Form15Misce()
    {
        InitializeComponent();
    }

    private void Form15Misce_Load(object sender, EventArgs e)
    {
        // TODO: This line of code loads data into the 'galereyaDataSet.misce' table.
        You can move, or remove it, as needed.

        this.misceTableAdapter.Fill(this.galereyaDataSet.misce);
    }

    private void button1_Click(object sender, EventArgs e)
    {
        //Створити рядок
        this.misceTableAdapter.Insert("введіть", 1, 0);
        //внести в базу зміни
        misceTableAdapter.Update(galereyaDataSet);
        //оновити дані в формі
        this.misceTableAdapter.Fill(this.galereyaDataSet.misce);
        int row1 = dataGridView1.RowCount - 1;
        if (row1 > -1)
        {
            dataGridView1.FirstDisplayedScrollingRowIndex = row1;
            DataGridViewCell cell = dataGridView1.Rows[row1].Cells[1];
            dataGridView1.CurrentCell = cell;
            dataGridView1.CurrentCell.Selected = true;
        }
    }
}

```

```

private void button2_Click(object sender, EventArgs e)
{
    //видалити рядок
    if (dataGridView1.SelectedRows.Count < 1)
    {
        MessageBox.Show("Не обрані рядки для видалення.");
    }
    else
    {
        //видалити всі виділені рядки
        foreach (DataGridViewRow row in dataGridView1.SelectedRows)
        {
            if (!row.IsNewRow) dataGridView1.Rows.Remove(row);
        }
    }
}

private void button3_Click(object sender, EventArgs e)
{
    //Зберегти зміни
    this.misceTableAdapter.Update(this.galereyaDataSet.misce);
}
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;

```

```

using System.Linq;
using System.Text;
using System.Windows.Forms;
// для бази даних
using System.Configuration;
using System.Data.Odbc;
using System.Globalization;

namespace galereya
{
    public partial class Form16Z_ekspoziciya : Form
    {
        public Form16Z_ekspoziciya()
        {
            InitializeComponent();
        }

        private void Form16Z_ekspoziciya_Load(object sender, EventArgs e)
        {

        }

        public void pokaz()
        {
            //Формувати
            // рядок підключення до бази даних
            string connectionString =
            ConfigurationManager.ConnectionStrings["galereya.Properties.Settings.galereyaC
onnectionString"].ConnectionString;

            OdbcConnection conn = new OdbcConnection(connectionString);

```

```

conn.Open();
OdbcCommand myCommand = conn.CreateCommand();
// запит
myCommand.CommandText = "SELECT DISTINCT m.misce_name, " +
    " COUNT(1) as kilkist, " +
    " COUNT(DISTINCT k.avtor_nrec) as k_avtor, " +
    " COUNT(DISTINCT k.janr_nrec) as k_janr, " +
    " SUM(k.cina) as varartist " +
    " FROM galereya2.misce m, galereya2.ruh h, galereya2.kartina k " +
    " WHERE h.d_end IS NULL " +
    " and m.pokaz = 1 " +
    " and h.misce_nrec=m.nrec " +
    " and h.kartina_nrec=k.nrec " +
    " GROUP BY m.misce_name " +
    " ORDER BY m.misce_name;";
// виконання запиту в новому адаптері
using (OdbcDataAdapter adapter = new OdbcDataAdapter())
{
    DataTable dt = new DataTable();
    adapter.SelectCommand = myCommand;
    //оновити дані в таблиці
    adapter.Fill(dt);
    dataGridView1.Columns.Clear();
    dataGridView1.DataSource = dt;
}
//закрити підключення до БД
conn.Close();
//заголовки
if (dataGridView1.ColumnCount == 5)
{

```

```

        dataGridView1.Columns[0].HeaderText = "Приміщення";
        dataGridView1.Columns[1].HeaderText = "Кількість картин";
        dataGridView1.Columns[2].HeaderText = "Кількість авторів";
        dataGridView1.Columns[3].HeaderText = "Кількість жанрів";
        dataGridView1.Columns[4].HeaderText = "Вартість, грн";
        dataGridView1.Columns[1].Width = 150;
        dataGridView1.Columns[2].Width = 150;
        dataGridView1.Columns[3].Width = 150;
        dataGridView1.Columns[4].Width = 150;
    }
}
}
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
// для бази даних
using System.Configuration;
using System.Data.Odbc;
using System.Globalization;

namespace galereya
{
    public partial class Form17Z_prodano : Form
    {

```

```

public Form17Z_prodano()
{
    InitializeComponent();
}

private void Form17Z_prodano_Load(object sender, EventArgs e)
{

}

public void sogodni()
{
    //сьогодні
    dateTimePicker1.Value = DateTime.Now;
    dateTimePicker2.Value = DateTime.Now;
}

public void pokaz()
{
    //Формувати
    string umova = " and (h.d_begin BETWEEN " +
        " CAST(" + String.Format("{0:yyyyMMdd}", dateTimePicker1.Value) + "
AS datetime) AND " +
        " CAST(" + String.Format("{0:yyyyMMdd}", dateTimePicker2.Value) + "
AS datetime) )";

    // рядок підключення до бази даних
    string connectionString =
        ConfigurationManager.ConnectionStrings["galereya.Properties.Settings.galereyaC
onnectionString"].ConnectionString;

    OdbcConnection conn = new OdbcConnection(connectionString);

```

```

conn.Open();
OdbcCommand myCommand = conn.CreateCommand();
// запит
myCommand.CommandText = "SELECT DISTINCT a.avtor_pib,
n.kraina_name, " +
    " COUNT(1) as kilkist, " +
    " SUM(k.cina) as varartist " +
    " FROM galereya2.misce m, galereya2.ruh h, galereya2.kartina k, " +
    " galereya2.avtor a, galereya2.kraina n" +
    " WHERE h.d_end IS NULL " + umova +
    " and m.nrec = 10 " +
    " and h.misce_nrec=m.nrec " +
    " and h.kartina_nrec=k.nrec " +
    " and k.avtor_nrec=a.nrec " +
    " and a.kraina_nrec=n.nrec " +
    " GROUP BY a.avtor_pib, n.kraina_name " +
    " ORDER BY a.avtor_pib, n.kraina_name ";
// виконання запиту в новому адаптері
using (OdbcDataAdapter adapter = new OdbcDataAdapter())
{
    DataTable dt = new DataTable();
    adapter.SelectCommand = myCommand;
    //оновити дані в таблиці
    adapter.Fill(dt);
    dataGridView1.Columns.Clear();
    dataGridView1.DataSource = dt;
}
//закрити підключення до БД
conn.Close();
//заголовки

```

					ДП ІС-5104.1260-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		135


```

if (dataGridView1.ColumnCount == 4)
{
    dataGridView1.Columns[0].HeaderText = "Автор";
    dataGridView1.Columns[1].HeaderText = "Країна";
    dataGridView1.Columns[2].HeaderText = "Кількість картин";
    dataGridView1.Columns[3].HeaderText = "Вартість, грн";
    dataGridView1.Columns[0].Width = 150;
    dataGridView1.Columns[1].Width = 150;
    dataGridView1.Columns[2].Width = 150;
    dataGridView1.Columns[3].Width = 150;
}
}

```

```

private void button1_Click(object sender, EventArgs e)
{
    //кнопка //Формувати
    pokaz();
}
}
}

```

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

// для бази даних
using System.Configuration;

```

					ДП ІС-5104.1260-с.ПЗ	Арк.
						136
Змн.	Арк.	№ докум.	Підпис	Дата		

```
using System.Data.Odbc;
using System.Globalization;
```

```
namespace galereya
```

```
{
```

```
public partial class Form18Z_vartist : Form
```

```
{
```

```
public Form18Z_vartist()
```

```
{
```

```
InitializeComponent();
```

```
}
```

```
private void Form18Z_vartist_Load(object sender, EventArgs e)
```

```
{
```

```
}
```

```
public void pokaz()
```

```
{
```

```
//Формувати
```

```
// рядок підключення до бази даних
```

```
string connectionString =
```

```
ConfigurationManager.ConnectionStrings["galereya.Properties.Settings.galereyaC
onnectionString"].ConnectionString;
```

```
OdbcConnection conn = new OdbcConnection(connectionString);
```

```
conn.Open();
```

```
OdbcCommand myCommand = conn.CreateCommand();
```

```
// запит
```

```
myCommand.CommandText = "SELECT DISTINCT m.misce_name, " +
```

```
" CASE m.pokaz WHEN 1 THEN 'в експозиції' ELSE 'зберігається' END
```

```
as pokaz_name, "+
```

					ДП ІС-5104.1260-с.ПЗ	Арк. 137
Змн.	Арк.	№ докум.	Підпис	Дата		

```

" COUNT(1) as kilkist, " +
" SUM(k.cina) as varartist " +
" FROM galereya2.misce m, galereya2.ruh h, galereya2.kartina k " +
" WHERE h.d_end IS NULL " +
" and m.nayavnist = 1 " +
" and h.misce_nrec=m.nrec " +
" and h.kartina_nrec=k.nrec " +
" GROUP BY m.misce_name, m.pokaz " +
" ORDER BY m.misce_name;";

```

// виконання запиту в новому адаптері

```
using (OdbcDataAdapter adapter = new OdbcDataAdapter())
```

```
{
```

```
    DataTable dt = new DataTable();
```

```
    adapter.SelectCommand = myCommand;
```

```
    //оновити дані в таблиці
```

```
    adapter.Fill(dt);
```

```
    dataGridView1.Columns.Clear();
```

```
    dataGridView1.DataSource = dt;
```

```
}
```

```
//закрити підключення до БД
```

```
conn.Close();
```

```
//заголовки
```

```
if (dataGridView1.ColumnCount == 4)
```

```
{
```

```
    dataGridView1.Columns[0].HeaderText = "Приміщення";
```

```
    dataGridView1.Columns[1].HeaderText = "Показ";
```

```
    dataGridView1.Columns[2].HeaderText = "Кількість картин";
```

```
    dataGridView1.Columns[3].HeaderText = "Вартість, грн";
```

```
    dataGridView1.Columns[1].Width = 150;
```

```
    dataGridView1.Columns[2].Width = 150;
```

					ДП ІС-5104.1260-с.ПЗ	Арк.
						138
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        dataGridView1.Columns[3].Width = 150;
    }
}
}
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
// для бази даних
using System.Configuration;
using System.Data.Odbc;
using System.Globalization;

namespace galereya
{
    public partial class Form19Z_kilkist_kraina : Form
    {
        public Form19Z_kilkist_kraina()
        {
            InitializeComponent();
        }

        private void Form19Z_kilkist_kraina_Load(object sender, EventArgs e)
        {

```

Змн.	Арк.	№ докум.	Підпис	Дата

}

public void pokaz()

{

//Формувати

// рядок підключення до бази даних

string connectionString =

ConfigurationManager.ConnectionStrings["galereya.Properties.Settings.galereyaC
onnectionString"].ConnectionString;

OdbcConnection conn = new OdbcConnection(connectionString);

conn.Open();

OdbcCommand myCommand = conn.CreateCommand();

// запит

myCommand.CommandText = "SELECT DISTINCT n.kraina_name,
m.misce_name, " +

" COUNT(1) as kilkist, " +

" SUM(k.cina) as vartist " +

" FROM galereya2.misce m, galereya2.ruh h, galereya2.kartina k, " +

" galereya2.avtor a, galereya2.kraina n" +

" WHERE h.d_end IS NULL " +

" and m.nayavnist = 1 " +

" and h.misce_nrec=m.nrec " +

" and h.kartina_nrec=k.nrec " +

" and k.avtor_nrec=a.nrec " +

" and a.kraina_nrec=n.nrec " +

" GROUP BY n.kraina_name, m.misce_name " +

" ORDER BY n.kraina_name, m.misce_name;";

// виконання запиту в новому адаптері

using (OdbcDataAdapter adapter = new OdbcDataAdapter())

{

					ДП ІС-5104.1260-с.ПЗ	Арк.
						140
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        DataTable dt = new DataTable();
        adapter.SelectCommand = myCommand;
        //оновити дані в таблиці
        adapter.Fill(dt);
        dataGridView1.Columns.Clear();
        dataGridView1.DataSource = dt;
    }
    //закрити підключення до БД
    conn.Close();
    //заголовки
    if (dataGridView1.ColumnCount == 4)
    {
        dataGridView1.Columns[0].HeaderText = "Країна";
        dataGridView1.Columns[1].HeaderText = "Приміщення";
        dataGridView1.Columns[2].HeaderText = "Кількість картин";
        dataGridView1.Columns[3].HeaderText = "Вартість, грн";
        dataGridView1.Columns[0].Width = 150;
        dataGridView1.Columns[1].Width = 150;
        dataGridView1.Columns[2].Width = 150;
        dataGridView1.Columns[3].Width = 150;
    }
}
}
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;

```

					ДП ІС-5104.1260-с.ПЗ	Арк.
						141
Змн.	Арк.	№ докум.	Підпис	Дата		

```

using System.Text;
using System.Windows.Forms;
// для бази даних
using System.Configuration;
using System.Data.Odbc;
using System.Globalization;

namespace galereya
{
    public partial class Form20Z_kilkist_avtor : Form
    {
        public Form20Z_kilkist_avtor()
        {
            InitializeComponent();
        }

        private void Form20Z_kilkist_avtor_Load(object sender, EventArgs e)
        {
        }

        public void pokaz()
        {
            //Формувати
            // рядок підключення до бази даних
            string connectionString =
            ConfigurationManager.ConnectionStrings["galereya.Properties.Settings.galereyaC
            onnectionString"].ConnectionString;

            OdbcConnection conn = new OdbcConnection(connectionString);
            conn.Open();

```

```

OdbcCommand myCommand = conn.CreateCommand();

// запит
myCommand.CommandText = "SELECT DISTINCT a.avtor_pib,
n.kraina_name, m.misce_name, " +
    " COUNT(1) as kilkist, " +
    " SUM(k.cina) as varartist " +
    " FROM galereya2.misce m, galereya2.ruh h, galereya2.kartina k, " +
    " galereya2.avtor a, galereya2.kraina n" +
    " WHERE h.d_end IS NULL " +
    " and m.nayavnist = 1 " +
    " and h.misce_nrec=m.nrec " +
    " and h.kartina_nrec=k.nrec " +
    " and k.avtor_nrec=a.nrec " +
    " and a.kraina_nrec=n.nrec " +
    " GROUP BY a.avtor_pib, n.kraina_name, m.misce_name " +
    " ORDER BY a.avtor_pib, n.kraina_name, m.misce_name;";

// виконання запиту в новому адаптері
using (OdbcDataAdapter adapter = new OdbcDataAdapter())
{
    DataTable dt = new DataTable();
    adapter.SelectCommand = myCommand;
    //оновити дані в таблиці
    adapter.Fill(dt);
    dataGridView1.Columns.Clear();
    dataGridView1.DataSource = dt;
}

//закрити підключення до БД
conn.Close();

//заголовки
if (dataGridView1.ColumnCount == 5)

```

					ДП ІС-5104.1260-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		143


```

    {
        dataGridView1.Columns[0].HeaderText = "Автор";
        dataGridView1.Columns[1].HeaderText = "Країна";
        dataGridView1.Columns[2].HeaderText = "Приміщення";
        dataGridView1.Columns[3].HeaderText = "Кількість картин";
        dataGridView1.Columns[4].HeaderText = "Вартість, грн";
        dataGridView1.Columns[0].Width = 150;
        dataGridView1.Columns[1].Width = 150;
        dataGridView1.Columns[2].Width = 100;
        dataGridView1.Columns[3].Width = 150;
        dataGridView1.Columns[4].Width = 150;
    }
}
}
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace galereya
{
    public partial class Form2Login : Form
    {
        public Form2Login()
        {

```

```

InitializeComponent();
}

private void button1_Click(object sender, EventArgs e)
{
    //кнопка Увійти
    this.DialogResult = DialogResult.OK;
}

private void button2_Click(object sender, EventArgs e)
{
    //кнопка Відміна
    this.DialogResult = DialogResult.Cancel;
}

public string get_login()
{
    //відкритий метод для доступу до логіна
    return (this.textBox1.Text.Trim());
}

public string get_pass()
{
    //відкритий метод для доступу до пароля
    return (this.textBox2.Text.Trim());
}
}

using System;

```

					ДП ІС-5104.1260-с.ПЗ	Арк.
						145
Змн.	Арк.	№ докум.	Підпис	Дата		

```

using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
// для бази даних
using System.Configuration;
using System.Data.Odbc;

namespace galereya
{
    public partial class Form3User : Form
    {
        public string connectionString; // рядок підключення до бази даних

        public Form3User()
        {
            InitializeComponent();
        }

        private void Form3User_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the 'galereyaDataSet.rol' table. You
            can move, or remove it, as needed.

            this.rolTableAdapter.Fill(this.galereyaDataSet.rol);

            // TODO: This line of code loads data into the 'galereyaDataSet.myuser' table.
            You can move, or remove it, as needed.

            this.myuserTableAdapter.Fill(this.galereyaDataSet.myuser);

```

					ДП ІС-5104.1260-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		146

}

```
private void button3_Click(object sender, EventArgs e)
```

{

```
//Зберегти зміни
```

```
this.myuserTableAdapter.Update(this.galereyaDataSet.myuser);
```

}

```
private void button4_Click(object sender, EventArgs e)
```

{

```
//Змінити пароль
```

```
if (dataGridView1.RowCount < 1)
```

{

```
MessageBox.Show("Оберіть користувача в таблиці");
```

}

```
string s1 = textBox1.Text;
```

```
string s2 = textBox2.Text;
```

```
int nrec = Int32.Parse(dataGridView1.CurrentRow.Cells[0].Value.ToString());
```

```
int row1 = dataGridView1.SelectedCells[0].RowIndex;
```

```
if (s1.Equals(s2) && nrec > 0)
```

{

```
//внести в базу зміни
```

```
myuserTableAdapter.Update(galereyaDataSet);
```

```
OdbcConnection conn = new OdbcConnection(connectionString);
```

```
conn.Open();
```

```
OdbcCommand myCommand = conn.CreateCommand();
```

```
// запит
```

```
myCommand.CommandText = "UPDATE [galereya2].[myuser] " +
```

```
" SET pass = galereya2.md5(?) WHERE nrec=? ;";
```

Змн.	Арк.	№ докум.	Підпис	Дата

```

// параметри запита
myCommand.Parameters.AddWithValue("@pass1", s1);
myCommand.Parameters.AddWithValue("@nrec1", nrec);
// виконати запит
myCommand.ExecuteNonQuery();
//оновити дані в формі
this.myuserTableAdapter.Fill(this.galereyaDataSet.myuser);
dataGridView1.FirstDisplayedScrollingRowIndex = row1;
DataGridViewCell cell = dataGridView1.Rows[row1].Cells[1];
dataGridView1.CurrentCell = cell;
dataGridView1.CurrentCell.Selected = true;
MessageBox.Show("Пароль змінено.");
}
else
{
    MessageBox.Show("Пароль і його повтор різні");
}
}

private void button1_Click(object sender, EventArgs e)
{
    //Створити користувача
    int rol_nrec = 4; //право на перегляд
    this.myuserTableAdapter.Insert(rol_nrec,
        DateTime.Today, DateTime.Today, "ПІБ", "введіть", "введіть");
    //внести в базу зміни
    myuserTableAdapter.Update(galereyaDataSet);
    //оновити дані в формі
    this.myuserTableAdapter.Fill(this.galereyaDataSet.myuser);
    int row1 = dataGridView1.RowCount - 1;

```

```
if (row1 > -1)
{
    dataGridView1.FirstDisplayedScrollingRowIndex = row1;
    DataGridViewCell cell = dataGridView1.Rows[row1].Cells[1];
    dataGridView1.CurrentCell = cell;
    dataGridView1.CurrentCell.Selected = true;
}
}
```

```
private void button2_Click(object sender, EventArgs e)
{
    //видалити користувача
    if (dataGridView1.SelectedRows.Count < 1)
    {
        MessageBox.Show("Не обрані рядки для видалення.");
    }
    else
    {
        //видалити всі виділені рядки
        foreach (DataGridViewRow row in dataGridView1.SelectedRows)
        {
            if (!row.IsNewRow) dataGridView1.Rows.Remove(row);
        }
    }
}

using System;
using System.Collections.Generic;
```

```
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
```

```
namespace galereya
```

```
{
```

```
public partial class Form4Filtr : Form
```

```
{
```

```
public Form4Filtr()
```

```
{
```

```
InitializeComponent();
```

```
}
```

```
private void button1_Click(object sender, EventArgs e)
```

```
{
```

```
//кнопка ОК
```

```
this.DialogResult = DialogResult.OK;
```

```
}
```

```
private void button2_Click(object sender, EventArgs e)
```

```
{
```

```
//кнопка Відміна
```

```
this.DialogResult = DialogResult.Cancel;
```

```
}
```

```
private void Form4Filter_Load(object sender, EventArgs e)
```

```
{
```

// TODO: This line of code loads data into the 'galereyaDataSet.p_misce' table.
You can move, or remove it, as needed.

```
this.p_misceTableAdapter.Fill(this.galereyaDataSet.p_misce);
```

// TODO: This line of code loads data into the 'galereyaDataSet.p_avtor' table.
You can move, or remove it, as needed.

```
this.p_avtorTableAdapter.Fill(this.galereyaDataSet.p_avtor);
```

// TODO: This line of code loads data into the 'galereyaDataSet.p_kraina' table.
You can move, or remove it, as needed.

```
this.p_krainaTableAdapter.Fill(this.galereyaDataSet.p_kraina);
```

// TODO: This line of code loads data into the 'galereyaDataSet.p_osnova' table. You can move, or remove it, as needed.

```
this.p_osnovaTableAdapter.Fill(this.galereyaDataSet.p_osnova);
```

// TODO: This line of code loads data into the 'galereyaDataSet.p_robota' table.
You can move, or remove it, as needed.

```
this.p_robotaTableAdapter.Fill(this.galereyaDataSet.p_robota);
```

// TODO: This line of code loads data into the 'galereyaDataSet.p_tehnika' table. You can move, or remove it, as needed.

```
this.p_tehnikaTableAdapter.Fill(this.galereyaDataSet.p_tehnika);
```

// TODO: This line of code loads data into the 'galereyaDataSet.p_janr' table.
You can move, or remove it, as needed.

```
this.p_janrTableAdapter.Fill(this.galereyaDataSet.p_janr);
```

```
}
```

```
public int get_janr()
```

```
{
```

```
//метод отримання вибору користувача - жанр
```

```
return Convert.ToInt32(comboBox1.SelectedValue);
```

```
}
```

```
public int get_tehnika()
```



```
{
    //метод отримання вибору користувача - техніка
    return Convert.ToInt32(comboBox2.SelectedValue);
}
```

```
public int get_robota()
{
    //метод отримання вибору користувача - робота
    return Convert.ToInt32(comboBox3.SelectedValue);
}
```

```
public int get_osnova()
{
    //метод отримання вибору користувача - основа
    return Convert.ToInt32(comboBox4.SelectedValue);
}
```

```
public int get_kraina()
{
    //метод отримання вибору користувача - країна
    return Convert.ToInt32(comboBox5.SelectedValue);
}
```

```
public int get_avtor()
{
    //метод отримання вибору користувача - автор
    return Convert.ToInt32(comboBox6.SelectedValue);
}
```

```
public int get_misce()
```

```
{  
    //метод отримання вибору користувача - місце знаходження  
    return Convert.ToInt32(comboBox7.SelectedValue);  
}  
  
public string get_nazva()  
{  
    //метод отримання вибору користувача - частина назви  
    return textBox1.Text.Trim();  
}  
  
public int get_rik()  
{  
    //метод отримання вибору користувача - рік  
    int rik = 0;  
    string s1 = textBox2.Text.Trim();  
    if (s1.Length>0 ) rik = Convert.ToInt32(s1);  
    return rik;  
}  
  
}  
}  
  
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Windows.Forms;
```

```
namespace galereya
{
    public partial class Form5About : Form
    {
        public Form5About()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            //кнопка ОК
            this.DialogResult = DialogResult.OK;
        }

        private void Form5About_Load(object sender, EventArgs e)
        {
        }
    }
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
```

```

namespace galereya
{
    public partial class Form6Edit : Form
    {
        public int nrec ; //ідентифікатор рядка або 0, якщо новий запис
        public DataGridViewRow r1; //старі дані рядка
        private bool is_first; //перший показ

        public Form6Edit()
        {
            InitializeComponent();
            is_first = true;
        }

        private void button1_Click(object sender, EventArgs e)
        {
            //кнопка ОК
            this.DialogResult = DialogResult.OK;
        }

        private void button2_Click(object sender, EventArgs e)
        {
            //кнопка Відміна
            this.DialogResult = DialogResult.Cancel;
        }

        private void Form6Edit_Load(object sender, EventArgs e)
        {

```

					ДП ІС-5104.1260-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		155

// TODO: This line of code loads data into the 'galereyaDataSet.misce' table.

You can move, or remove it, as needed.

```
this.misceTableAdapter.Fill(this.galereyaDataSet.misce);
```

// TODO: This line of code loads data into the 'galereyaDataSet.avtor' table.

You can move, or remove it, as needed.

```
this.avtorTableAdapter.Fill(this.galereyaDataSet.avtor);
```

// TODO: This line of code loads data into the 'galereyaDataSet.osnova' table.

You can move, or remove it, as needed.

```
this.osnovaTableAdapter.Fill(this.galereyaDataSet.osnova);
```

// TODO: This line of code loads data into the 'galereyaDataSet.robota' table.

You can move, or remove it, as needed.

```
this.robotaTableAdapter.Fill(this.galereyaDataSet.robota);
```

// TODO: This line of code loads data into the 'galereyaDataSet.tehnika' table.

You can move, or remove it, as needed.

```
this.tehnikaTableAdapter.Fill(this.galereyaDataSet.tehnika);
```

// TODO: This line of code loads data into the 'galereyaDataSet.janr' table. You can move, or remove it, as needed.

```
this.janrTableAdapter.Fill(this.galereyaDataSet.janr);
```

```
}
```

```
private void Form6Edit_Activated(object sender, EventArgs e)
```

```
{
```

```
// активація форми
```

```
if (is_first && nrec > 0)
```

```
{
```

```
//перший показ редагування - показати дані обраного рядка
```

```
comboBox1.SelectedValue = Convert.ToInt32(r1.Cells[1].Value);
```

```
comboBox2.SelectedValue = Convert.ToInt32(r1.Cells[2].Value);
```

```
comboBox3.SelectedValue = Convert.ToInt32(r1.Cells[3].Value);
```

```

comboBox4.SelectedValue = Convert.ToInt32(r1.Cells[4].Value);
comboBox5.SelectedValue = Convert.ToInt32(r1.Cells[5].Value);
textBox1.Text = r1.Cells[6].Value.ToString();
textBox3.Text = r1.Cells[7].Value.ToString();
textBox4.Text = r1.Cells[8].Value.ToString();
textBox2.Text = r1.Cells[11].Value.ToString();
textBox5.Text = r1.Cells[12].Value.ToString();
textBox6.Text = r1.Cells[13].Value.ToString();
textBox7.Text = r1.Cells[20].Value.ToString();
//сховати місце
label13.Visible = false;
comboBox6.Visible = false;
is_first = false; //щоб більше не запускати цей блок
    }
}

```

```

    }
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

```

```
namespace galereya
```

```
{
```

```
    public partial class Form7Ruh : Form
```

					ДП ІС-5104.1260-с.ПЗ	Арк. 157
Змн.	Арк.	№ докум.	Підпис	Дата		

```

{
    public Form7Ruh()
    {
        InitializeComponent();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        //кнопка ОК
        this.DialogResult = DialogResult.OK;
    }

    private void button2_Click(object sender, EventArgs e)
    {
        //кнопка Відміна
        this.DialogResult = DialogResult.Cancel;
    }

    private void Form7ruh_Load(object sender, EventArgs e)
    {
        // TODO: This line of code loads data into the 'galereyaDataSet.misce' table.
        You can move, or remove it, as needed.
        this.misceTableAdapter.Fill(this.galereyaDataSet.misce);
    }

    public int get_misce()
    {
        //метод отримання вибору користувача - місце знаходження
        return Convert.ToInt32(comboBox1.SelectedValue);
    }

```

```

    }
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;

namespace galereya
{
    public partial class Form8Foto : Form
    {
        private string dir_foto; // директорія фото
        private int kartina_nrec; // ідентифікатор картини
        private int user_rol; // роль користувача

        public Form8Foto()
        {
            InitializeComponent();
        }

        public void set_parametri(string dir_foto1, int kartina_nrec1, int user_rol1 )
        {
            // початкова настройка форми
            dir_foto = dir_foto1;

```

					ДП ІС-5104.1260-с.ПЗ	Арк.
						159
Змн.	Арк.	№ докум.	Підпис	Дата		


```

kartina_nrec = kartina_nrec1;
user_rol = user_rol1;
fotoBindingSource.Filter = string.Format("kartina_nrec = {0}", kartina_nrec);
//this.fotoTableAdapter.Fill(this.galereyaDataSet.foto);
if (user_rol > 3)
{
    //обмежити доступ до редагування
    button1.Enabled = false;
    button2.Enabled = false;
    button3.Enabled = false;
    dataGridView1.ReadOnly = true;
}
}

private void show_ris( string fn )
{
    //показ файла
    if (System.IO.File.Exists(fn))
    {
        System.IO.FileStream fs = new System.IO.FileStream(fn,
System.IO.FileMode.Open);
        System.Drawing.Image img = System.Drawing.Image.FromStream(fs);
        fs.Close();
        pictureBox1.Image = img;
    }
}

private void Form8Foto_Load(object sender, EventArgs e)
{

```

// TODO: This line of code loads data into the 'galereyaDataSet.foto' table. You can move, or remove it, as needed.

```
this.fotoTableAdapter.Fill(this.galereyaDataSet.foto);
}
```

```
private void button1_Click(object sender, EventArgs e)
{
```

```
//Створити
```

```
string fn1 = "", fn2 = "", fn0 = "";
```

```
if (openFileDialog1.ShowDialog() == DialogResult.OK)
```

```
{
```

```
    fn1 = openFileDialog1.FileName;
```

```
}
```

```
else
```

```
{
```

```
    return;
```

```
}
```

```
//сформувати новий шлях, щоб не співпав з наявними файлами
```

```
FileInfo test = new FileInfo( fn1 );
```

```
string name = test.Name; // імя файла
```

```
string ext = test.Extension; // тип файла
```

```
int i = 0;
```

```
name = name.Substring(0, name.Length - ext.Length );
```

```
fn0 = name + ext;
```

```
fn2 = dir_foto + fn0;
```

```
while (System.IO.File.Exists(fn2))
```

```
{
```

```
    i++;
```

```
    fn0 = name + i.ToString() + ext;
```

```
    fn2 = dir_foto + fn0;
```

					ДП ІС-5104.1260-с.ПЗ	Арк.
						161
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }

    //копіювання файла
    File.Copy(fn1, fn2);

    //вставка рядка
    this.fotoTableAdapter.Insert(kartina_nrec, fn0, "");

    //внести в базу зміни
    fotoTableAdapter.Update(galereyaDataSet);

    //оновити дані в формі
    this.fotoTableAdapter.Fill(this.galereyaDataSet.foto);

    int row1 = dataGridView1.RowCount - 1;
    if (row1 > -1)
    {
        dataGridView1.FirstDisplayedScrollingRowIndex = row1;
        DataGridViewCell cell = dataGridView1.Rows[row1].Cells[2];
        if (cell != null)
        {
            dataGridView1.CurrentCell = cell;
            dataGridView1.CurrentCell.Selected = true;
        }
    }
}

private void button2_Click(object sender, EventArgs e)
{
    //видалити
    if (dataGridView1.SelectedRows.Count < 1)
    {
        MessageBox.Show("Не обрані рядки для видалення.");
    }
    else

```

```

{
    string fn;
    //видалити всі виділені рядки
    foreach (DataGridViewRow row in dataGridView1.SelectedRows)
    {
        if (!row.IsNewRow)
        {
            //видалити файл
            fn = dir_foto + row.Cells[2].Value.ToString();
            try
            {
                File.Delete( fn );
                MessageBox.Show("Виделено файл " + fn);
            }
            catch
            {
                MessageBox.Show("Помилка виделення файла " + fn);
            }
            //видалити рядок
            dataGridView1.Rows.Remove(row);
        }
    }
}

private void button3_Click(object sender, EventArgs e)
{
    //Зберегти зміни
    this.fotoTableAdapter.Update(this.galereyaDataSet.foto);
}

```

					ДП ІС-5104.1260-с.ПЗ	Арк.
						163
Змн.	Арк.	№ докум.	Підпис	Дата		

```

private void dataGridView1_RowEnter(object sender,
DataGridViewCellEventArgs e)
{
    //змінився обраний рядок
    // ім'я файла
    string fn = dir_foto +
dataGridView1.Rows[e.RowIndex].Cells[2].Value.ToString();
    // показ файла
    show_ris( fn );
}

}
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace galereya
{
    public partial class Form9Janr : Form
    {
        public Form9Janr()
        {
            InitializeComponent();

```

					ДП ІС-5104.1260-с.ПЗ	Арк.
						164
Змн.	Арк.	№ докум.	Підпис	Дата		

}

```
private void button3_Click(object sender, EventArgs e)
```

{

```
//Зберегти зміни
```

```
this.janrTableAdapter.Update(this.galereyaDataSet.janr);
```

}

```
private void button2_Click(object sender, EventArgs e)
```

{

```
//видалити рядок
```

```
if (dataGridView1.SelectedRows.Count < 1)
```

{

```
    MessageBox.Show("Не обрані рядки для видалення.");
```

}

```
else
```

{

```
//видалити всі виділені рядки
```

```
foreach (DataGridViewRow row in dataGridView1.SelectedRows)
```

{

```
    if (!row.IsNewRow) dataGridView1.Rows.Remove(row);
```

}

}

}

```
private void button1_Click(object sender, EventArgs e)
```

{

```
//Створити рядок
```

```
this.janrTableAdapter.Insert("введіть");
```

```
//внести в базу зміни
```

					ДП ІС-5104.1260-с.ПЗ	Арк.
						165
Змн.	Арк.	№ докум.	Підпис	Дата		

```

janrTableAdapter.Update(galereyaDataSet);
//оновити дані в формі
this.janrTableAdapter.Fill(this.galereyaDataSet.janr);
int row1 = dataGridView1.RowCount - 1;
if (row1 > -1)
{
    dataGridView1.FirstDisplayedScrollingRowIndex = row1;
    DataGridViewCell cell = dataGridView1.Rows[row1].Cells[1];
    dataGridView1.CurrentCell = cell;
    dataGridView1.CurrentCell.Selected = true;
}
}

private void Form9Janr_Load(object sender, EventArgs e)
{
    // TODO: This line of code loads data into the 'galereyaDataSet.janr' table. You
    can move, or remove it, as needed.
    this.janrTableAdapter.Fill(this.galereyaDataSet.janr);
}
}
}

```